AD-A056 509 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 6/4
AUTOMATIC RECOGNITION OF SYNTHETIC SPEECH USING AN ELECTRONIC M--ETC(U)
JUN 78 D B WARMUTH

UNCLASSIFIED AFIT/DS/EE/78-3 NL

1 OF 2
AD
A056 509

14

AFIT/DS/EE/78-3

LEVEL II

16 7233

17 Ø3

6

AUTOMATIC RECOGNITION OF SYNTHETIC
SPEECH USING AN ELECTRONIC MODEL
OF THE MIDDLE AND INNER EAR

DISSERTATION

AFIT/DS/EE/78-3 10 Donald B. Warmuth
Capt USAF

9 Doctoral thesis

11 5 Jun 78

12 168 p.

Approved for public release; distribution unlimited

Ø12 225

78 07 07 009

AFIT/DS/EE/78-3

AUTOMATIC RECOGNITION OF SYNTHETIC SPEECH USING AN ELECTRONIC MODEL OF THE MIDDLE AND INNER EAR

DISSERTATION

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

by

Donald B. Warmuth, B.S., M.S.
Captain USAF

AFIT/DS/EE/78-3

AUTOMATIC RECOGNITION OF SYNTHETIC SPEECH USING AN ELECTRONIC MODEL OF THE MIDDLE AND INNER EAR

by

Donald B. Warmuth, B.S., M.S.

Captain USAF

Approved:

______________________ 5 June 1978
Chairman

______________________ 5 June 1978

______________________ 5 June 1978

Thomas E. Reeves 5 June 1978

______________________ 5 June 1978

Accepted:

______________________ 5 June 1978
Dean, School of Engineering

Preface

This dissertation is the result of an attempt to recognize the output of the speech synthesis system developed as a part of my Master Degree program. It is proposed as a starting point for the future development of an automatic speech recognition system for natural speech. We have shown that the ROC COC Filter and the CxC Computer are a viable feature extraction system for the analysis of speech and possibly other audio frequency signals. If you need to use or maintain, or hopefully, enhance the computer programs which were developed, I call your attention to Appendix B.

This dissertation was written for a technically-oriented individual who has little or no knowledge of speech generation, speech synthesis, hearing, or speech recognition. Should this dissertation whet the reader's appetite for more information in these areas, there are several excellent books and articles listed in the bibliography.

I wish to acknowledge my indebtedness to Dr. J. Ryland Mundie of the Aerospace Medical Research Laboratories for having the time, patience, and understanding to help see me through this project. I would also like to acknowledge my indebtedness to my advisor, Dr. Gregg L. Vaughn, who came aboard in mid-stream and proved to be invaluable, to my father, Leo A. Warmuth, for his encouragement, and to an old friend, Dennis Kono, who provided an inspiration when one was desperately needed. A very special thanks has to go to my wife, Debbie, without whose courage, stamina, and prodding this project would never have been completed.

Contents

Contents

List of Figures

List of Figures

List of Figures

List of Tables

Abstract

A phoneme-based automatic speech recognition system was developed and tested using synthetic speech. The acoustic signal is divided into short segments for analysis; segments are either a single pitch period of voiced speech or a 10 ms sample of voiceless speech. These segments are independently analyzed and given a phonemic name by three different measures. The sub-phonemic segments are grouped using measures which reflect dynamic changes in the speech signal. Each group of segments represents a phoneme and is identified by simple algorithms operating on the string of phonemically named segments that form the group.

Although synthetic speech was used to develop and test the recognition system, classification was based only on features present in natural speech. The specific speech synthesis system used was developed by the author as a previous project and generates continuous speech from a string of phonemes as input. Thus, it was possible to directly compare the output of the phoneme recognizer with the input to the speech synthesizer.

The phoneme recognizer was built around two previously developed pieces of electronic hardware. The first, the ROC COC Filter, models the sound transformations of the middle and inner ear and is used as an acoustic signal pre-processor. It uses a band-pass filter to simulate the middle ear and a very unique electronic transmission line to simulate the hydro-mechanical functions of the inner ear. The second piece of hardware, the CxC Computer, models the logic responses of the nervous system and is used as a feature extractor. In combination these models produce a partial simulation of the neural impulses that have been detected at this level in animals in response to speech

sounds.

The phoneme recognition system was tested using isolated synthesized words which permitted evaluation with connected phoneme strings but stopped short of requiring development of word boundary rules. The tests consisted of 100 phonemically balanced words containing 281 phonemes. Of these, 245 phonemes were correctly identified, 23 were mis-identified, 13 were missed entirely, and 11 were added. However, many of the errors were predictable or understandable and may be overcome at a higher (word or phrase) level. It is firmly believed that with further research and the addition of some simple phonetic and linguistic rules this system can be developed into a working natural speech recognizer that requires only a small computer (or a small part of a large one), requires relatively small amounts of processing time, and has the potential of an almost unlimited vocabulary.

AUTOMATIC RECOGNITION OF SYNTHETIC SPEECH

USING AN ELECTRONIC MODEL OF THE MIDDLE AND INNER EAR

I. Introduction

Speech is a convenient and universal method of communication. Communication by speech requires three components, the speaker, the message, and the listener, all of which must be functioning. No matter how valiantly the speaker tries or how accurate the message, if the listener cannot recognize and understand the message no communication occurs. Because spoken communication is so fast and efficient, it is the primary means of conveying information from one person to another and has intrigued scientists for centuries (Ref. 5). They have studied this form of communication from the original thought processes through the generation of speech to the act of hearing and understanding. Currently, considerable effort is being expended on the analysis of the various aspects of hearing and on an attempt to simulate the hearing process using electronic and digital computer models. Scientists in the Aerospace Medical Research Laboratories (AMRL) have developed electronic models of the signal transformation functions of the middle and inner ear and the feature extraction functions of the lower auditory nervous system. The AMRL models are based on studies of the physical auditory system of animals, which are known to be capable of speech recognition as well as a variety of other tasks.

It is the purpose of this thesis to determine if automatic speech recognition is possible with the AMRL models. Inherent in this task are fundamental pattern recognition problems such as development of

methods of pattern measurement and classification and definition of pattern boundaries (segmentation on compartmentalization). In addition, rules will be developed for derivation of the basic units of speech (phonemes) from the results of the audio signal classification processes. Accomplishing this task will also demonstrate the AMRL models contain sufficient information through the signal transformation and feature extraction operations. If successful, the system might be developed into a useful analog signal pattern classifier for which automatic speech recognition would be only one important application.

Basic Terminology

A basic grasp of some of the terms used in the discussion of speech production and recognition is necessary to understand the research presented in this thesis.

A phoneme is a basic unit of spoken language. It is the smallest unit of language which, when exchanged for another such unit, will change the meaning of a word. Phonemes bear the same relationship to spoken language as alphabetic characters bear to written language. In English, 40 to 44 phonemes are generally recognized. Each of these may be represented by a written symbol, and several such symbol sets are in use. Table I on page 3 lists two of these symbol sets, the International Phonetic Alphabet (IPA) and the International Teaching Alphabet (ITA), as well as the teletype code used to represent the phonemes in this project. An example word is also listed for each phoneme.

TABLE I

DEFINITION OF SYMBOLS

Teletype Code	IPA Symbol	ITA Symbol	Typical Word
Vowels			
IY	i	ee	beet
II	I	i	bit
EE	ɛ	e	bet
AE	æ	a	bat
AA	ɑ	o	box
UH	ʌ	u	but
UU	U	ω	book
OO	u	ω	boot
OW	ɔ	au	bought
ER	ɝ	ɹ	bird
Semivowels			
WW	w	w	word
LL	l	l	love
RR	r	r	run
YY	y	y	yes
Voiced Stops			
BB	b	b	bat
DD	d	d	dog
GG	g	g	got
Voiceless Stops			
PP	p	p	pot
TT	t	t	tot
KK	k	k	cot
Nasals			
MM	m	m	mat
NN	n	n	nap
NG	ŋ	ŋ	sing
Voiced Fricatives			
VV	v	v	very
TE	ð	th	the
ZZ	z	z	zero
ZH	ʒ	ʒ	azure
Voiceless Fricatives			
FF	f	f	fine
TH	θ	th	thick
SS	s	s	say
SH	ʃ	ʃh	shoot
Aspirant			
HH	h	h	help
Affricates			
CH	tʃ	ch	church
JJ	dʒ	j	judge
Diphthongs			
EI	eI	æ	weigh
AI	aI	ie	tie
OI	ɔI	oi	toy
OU	oU	œ	toe
AU	aU	ou	out

Any one phoneme may be produced as several somewhat different sounds depending upon context or dialect. Each such variation of a phoneme is called an allophone of the phoneme.

When speech is analyzed with a specially designed spectral analyzer called a spectrograph (Ref. 7), spectral peaks appear. The spectral peaks are clearly visible in spectrograms (outputs of the spectrograph) such as the one shown in Figure 1 on page 5 . By extensive study of the spectrograms of known sounds, speech scientists have identified certain principle spectral peaks with certain sounds (Ref. 2). Generally three such peaks are identified in each sound and are called formants. The formant with the lowest frequency is referred to as the first formant and lies in the range of 200 to 800 Hz. The formant with the next higher frequency is referred to as the second formant and lies in the range of 700 to 2000 Hz. The next higher formant is the third formant and lies in the range of 1800 to 3500 Hz. As is evident from the above, the frequency regions of adjacent formants overlap. If the sound being analyzed is unknown to the interpretor of the spectrogram and a peak appears in the overlap of two regions, determination has to be made as to whether it is a principle peak and if so, which of the two possible formants it is. Currently, the only way to resolve this ambiguity is to know what phoneme was being produced. The determination of which spectral peaks are significant when the speech sound is unknown is so difficult that when highly qualified spectrogram readers were given spectrograms of English sentences they could not identify the phonemes which had made up the utterances (Ref. 6). The problem of formant identification has long been thought to be the key to speech

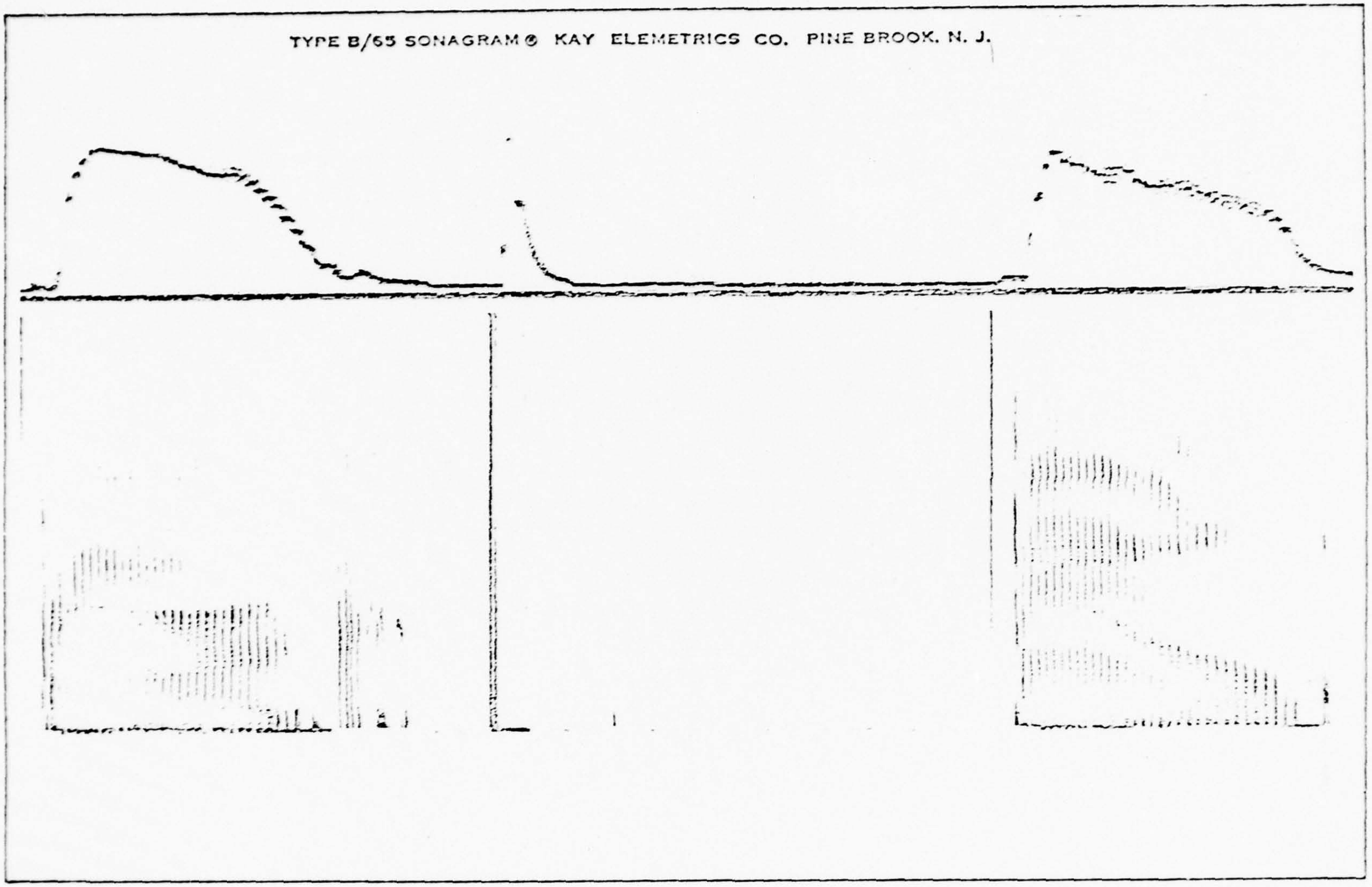


Fig. 1. Typical Spectrogram

recognition and several speech recognition systems have been based on this premise. However, they have met with only moderate success (References 4, 12, 13, and 18).

How Speech is Generated

The human vocal tract is a set of variable acoustic elements, under the control of the speaker. The vocal tract is excited by a periodic impulse source generated by motion of the vocal cords and/or by a noise resulting from a continuous turbulence near a constriction in the vocal tract. Speech sounds result as a modification of the source spectrum by the acoustic properties of the vocal tract elements. The various cavities (pharynx, mouth, nasal, between the teeth and lips) are acoustic elements and the moveable structures (lips, tongue, jaw, and velum) modify the elements producing different sounds. Sounds produced by vocal cord excitation are called voiced sounds. Sounds produced by a noise source are called unvoiced, voiceless, or fricative sounds. Sounds produced by the combination of vocal cord movement and noise are called voiced fricatives.

Sounds can be categorized by the method of excitation and by the site and extent of constriction of the vocal tract. With the vocal tract open and the vocal cords vibrating, the acoustic elements are excited by impulses of air released by the vocal cords and continuous voiced sounds (vowels, diphthongs, and semi-vowels) are produced by modifying the various acoustic elements. Vowels are sounds in which the formants approach and remain near some steady-state values. The diphthongs are sounds which start at one vowel and then proceed to or

toward another vowel. The semi-vowels are really consonants but are so named because of their vowel-like features.

When the vocal tract is constricted and the vocal cords are relaxed, a continuous turbulance is produced near the site of constriction which excites the acoustic elements to produce voiceless sounds. However, if the vocal cords also vibrate, the acoustic elements are excited by both turbulance and air impulses producing the voiced fricatives. Specific phonemes are produced by varying the place of constriction as well as the configuration of the acoustic elements. Closure of the teeth causes either a voiceless fricative (S or SH) or a voiced fricative (Z or ZH). Placing the tip of the tongue to the teeth causes either the voiceless TH or the voiced TE. Moving the lower lip to the teeth causes either the voiceless F or the voiced V. The special case in which the point of constriction is the glottis is called aspiration. In this instance the vocal cords are placed mid-way between the fully open position and the closed position. There is no vocal cord vibration, but there is turbulance. Constriction at the glottis is used for the phoneme H and for whispering voiced sounds.

Complete closure in the mouth and opening of the nasal cavity are used to generate the nasals. Air passage through the mouth is stopped by either the tongue or the lips and the velum drops to allow the air to pass through the nasal cavity. In nasals, as in fricatives, the closure location and the condition of the acoustic elements determine which phoneme is produced. Closure of the lips causes an M; closure by the tip of the tongue to the roof of the mouth causes an N; closure by the back of the tongue to the back of the mouth causes an NG.

Stops are also formed by a closure in the mouth but the air is not allowed to pass through the nasal cavity. Thus, the air flow through the vocal tract is completely stopped. However, the diaphram continues to move causing a pressure buildup behind the blockage. The blockage is removed suddenly causing a surge of air. Therefore, a stop can be characterized by a rapid closure, a short period of silence, and a rapid release. A stop is either voiced or voiceless depending on the condition of the vocal cords at the time of closure and release. In a voiced stop, voicing may precede or accompany the release. In a voiceless stop, voicing is delayed for 30-40 ms after the release resulting in a burst of fricative noise. In stops, as in fricatives and nasals, the phoneme produced is determined by the closure location and the condition of the acoustic elements. Closure at the lips is used for either a voiced B or a voiceless P; closure by the tip of the tongue to the roof of the mouth is used for a voiced D or a voiceless T; closure by the back of the tongue to the back of the mouth is used for a voiced G or a voiceless K.

Synthetic Speech Generation

Man has attempted to synthesize speech for centuries. These attempts have ranged from the use of bellows and levers in early models to the use of high-speed digital computers and filter systems. Reference 8 gives a history of speech synthesis. One goal of the research into speech synthesis was to make machines "talk"; but the research was also to increase understanding of speech production and recognition. In the last few years interest in speech synthesis has been aroused

because of the availability of high-speed digital computers and because of an increased understanding of the process of speech production. The main thrust of research in this area has been on speech synthesis by rule.

Speech synthesis by rule is the production of recognizable artificial speech in a given dialect by transforming a written representation of the utterance into a continuous waveform output. The written representation uses a set of symbols to represent phonemes, stress, pitch and pauses. Speech synthesis by rule normally requires two major components: a synthesizer which produces an analog output by modeling certain aspects of the vocal tract, and a synthesis strategy which controls the synthesizer. Some success has been achieved by Bell Telephone Laboratories (Ref. 1) by using the physical aspects of the vocal tract such as the masses and response times of the tongue, jaw, velum as the basis for synthesis strategy. Another approach has been to model the acoustic consequences of the various configurations of the vocal tract using measures from real speech waveforms to guide the synthesis. Two individuals have been relatively successful in the latter approach by treating the vocal tract as a parallel or cascaded set of complex pole and zero networks (Ref. 2:175-188). I. G. Mattingly (Ref. 8) took the parallel approach and L. R. Rabiner (Ref. 14) took the cascaded approach. There is controversy about which approach is better (Ref. 14:62 and Ref. 8:36), but it seems the parallel system makes vowel and vowel-like sounds easier, while the cascade system is more capable of producing fricatives and stops. The cascade synthesizer, which Rabiner developed as a software digital filter on a computer, was

produced in hardware by Rockland Systems Corporation under the name of Digital Speech Synthesizer Model 4516. This was the synthesizer used in this project. A concise explanation of the synthesis strategy used to control the synthesizer is presented in Appendix A.

Hearing

The peripheral auditory system (outer, middle, and inner ear) transforms an acoustic wave into neural pulses that are transmitted to the brain. How the brain interprets these inpulses is an enigma. This discussion will be limited to a very simplistic explanation of the transformation of acoustic waves into neural impulses.

The outer ear traps the acoustic wave and channels it to the middle ear. The middle ear acts as an impedance matching transformer that transitions the wave from the outside air to the fluid in the inner ear. The mechanical components of the middle ear have inherent characteristics such as mass, inertia, and elasticity which cause the sound signal to be band-pass filtered on its way to the inner ear.

The inner ear (cochlea) is a fluid-filled tube that contains a large number of detector cells and neurons. The cochlea is partitioned by a relatively thick membrance called the basilar membrane. The basilar membrane forms a surface for the fluid in the cochlea and transforms the longitudinal acoustic wave into a translational (surface) wave. The characteristics of the basilar membrane (such as size, thickness, and elasticity) vary dramatically along the length of the membrane. The results of this variation dominate the effects of the cochlea on the sound wave. The cochlea systematically reduces the

propagation velocity of the signal and systematically attentuates the signal as a function of distance as the signal passes down the cochlea.

Attached to the basilar membrane are approximately 10^4 sensory detector cells. These cells sense the displacement, velocity, or both of the membrane and drive the inputs of a network of neurons. The effect of the detector cells on the neurons is unknown; however, the neurons are known to exhibit changes in the rate they output impulses as a result of motion of the basilar membrane.

Synthetic Hearing (Automatic Speech Recognition)

Recent attempts at automatic speech recognition can be organized into three basic categories. The first, and most complex, is a "top-down" procedure. In this approach the presence of a particular word within a certain section of the utterance is predicted based on linguistic, semantic, syntactic, and phonetic rules. Several acoustic measures such as formant positions, formant trajectories, amplitude variations and voiced/voiceless determinations are hypothesized from the predicted word. A probability of occurrence of the predicted word is calculated by comparing the hypothesized measures with actual measures of the utterance section being analyzed. Such systems have been evaluated by their ability to properly respond to complete phrases or sentences and hence are called speech understanding systems.

The above research has four basic characteristics. The researchers are basing their systems on formant tracking; they are recognizing words from a given vocabulary; they are analyzing the

speech signal in constant length segments, typically 10 ms long; and they are using natural speech. Formant tracking is the monitoring of the time evolution of major peaks of the power spectrum of speech. In order to perform formant tracking, the formants (or at least a comparable measure) must be extracted from the speech signal. Many automatic speech recognition systems perform some sort of spectral analysis in the recognition process but analysis is sometimes done in terms of autocorrelations of the amplitude variations of the speech waveform, or in terms of linear predictive codes (Ref. 4), or in terms of zero crossing statistics (Ref. 11).

All current attempts at continuous speech recognition are top-down systems and they have not progressed beyond being "laboratory curiosities." The Advanced Research Projects Administration (ARPA) has sponsored a five-year speech-understanding project which has given a great deal of impetus to research in continuous speech recognition. Involved in this research are such prestigious facilities as Bolt, Beranek, and Newman; Carnegie-Mellon University; Lincoln Laboratory (MIT); Standford Research Institute; Systems Development Corporation; Haskins Laboratories; Speech Communication Research Laboratory; Sperry-Rand; and the University of California at Berkely. Commercially, Bell Telephone Laboratories, IBM, and Texas Instruments are also involved in speech recognition.

The second, and most simplistic, category of speech recognition systems is an isolated word recognizer. These systems operate on acoustic measures of a signal sample bounded by silence. They treat any sound preceded and followed by silence as a single pattern; this pattern may be a word or short phrase. Features are extracted from

the audio signal and compared to each candidate in a set of stored prototypes. The "closest" match is recognized as the word or phrase for that sample. Such a system is clearly limited to a small vocabulary since each sample must be tested against all prototypes. Performance of these systems is determined by the percentage of correctly identified words or phrases and depends to a large extent upon the acoustic dissimiliarity of the members of the prototype set. Devices in this category are commercially available (Ref. 19) and are finding limited applications.

The third category of speech recognition is the "bottom-up" approach. In this approach the audio signal is partitioned into basic speech units (phonemes). It is generally recognized that there are only 40-44 phonemes in the language. Therefore, a phonemic-based system requires only a few prototypes in order to recognize all words, phrases, and sentences. There are two major problems in developing such a system: several acoustic representations for a particular phoneme (allophones) must be considered and the speech signal must be partitioned into phonemic units. A phoneme recognizer should be evaluated by the percentage of phonemes correctly identified in connected speech. Reference 19 is a recent overview of the state of the art of speech recognition.

Approach

For this dissertation it was decided to attempt recognition of speech on a phoneme-by-phoneme basis. This approach was selected because recognition at the phoneme level requires only a small number of prototypes for an almost unlimited vocabulary. Further, it was

decided to use synthetic speech rather than natural speech for development and testing the recognition system. Synthetic speech eliminates several problems inherent in natural speech while preserving the major attributes. The exact configuration of synthetic speech is known, whereas in natural speech, the actual speech sounds being analyzed are not precisely known and can be estimated only by presentation to a panel of trained listeners. The speech synthesis system which was used was developed by the author (Ref. 17) as an MS(EE) thesis project sponsored by AMRL. This system generates continuous speech from a string of phonemes as an input. Thus, it was possible to directly compare the output of the phonome recognizer with the input to the speech synthesizer.

Certain bounds were placed on this dissertation. The phoneme recognizer was to use the signal transformation function of the AMRL electronic model of the middle and inner ear (ROC COC Filter) as a preprocessor. Feature extraction was to be performed by the AMRL model (CxC Computer) of the first level of the auditory neural net. In combination these models produce a partial simulation of the neural impulses that have been detected at this level in animals in response to speech sounds. Further, this equipment marks the approximate beginning of each pitch period during voiced sounds which produces segmentation of the speech signal into a natural periodicity. No parameters or characteristics known to be unique to the speech synthesizer were to be used in the audio signal classification process; classification was to be based only on features present in natural speech. Contextual information (i.e. syntactic, semantic, prosodic, or phonetic rules) was not to be used. The AMRL equipment is coupled to a

PDP-11/20 digital computer system through a special interface. Therefore, to avoid interfacing problems, it was decided to limit computational and storage capability to the PDP-11/20. All new computer programs developed were restricted to Fortran IV. Program documentation and alteration of Fortran IV outweighed the speed advantage of PDP-11 assembly language. Within these limitations, an accurate phoneme recognition system was developed which operates in approximately 1000 times real-time and has the potential of an almost unlimited word vocabulary.

II. The Synthetic Speech Recognition System

A block diagram of the synthetic speech recognition system developed in this dissertation is shown in Figure 2 on page 17. This diagram shows how synthetic speech is used to take a written representation of speech through an audio signal and back to a written representation. At a future date the two major components of this system may be reversed producing a system that will accept natural speech as an input, transmit the phonemic content of the speech, and produce reconstructed speech at the output. Transmitting speech in this matter will require a data rate of less than 100 bps. Speech transmission is, of course, only one of a great number of uses for automatic speech recognition (Ref. 16).

The synthetic speech recognition system was built around three specialized pieces of hardware. These are the speech synthesizer, the ROC COC Filter, and the CxC Computer which are briefly described in this chapter. Complete descriptions of the ROC COC Filter and the CxC Computer are available in References 14 and 10. A concise explanation of the synthesis strategy used to control the speech synthesizer is presented in Appendix A . However, for a general understanding of this work, the material presented here should be sufficient.

The Synthesizer

The Rockland Model 4516 Speech Synthesizer is a hardware version of a software speech synthesizer developed by L. R. Rabiner (Ref. 14). It models the acoustic consequences of the various configurations of the vocal tract. The transfer function of the vocal tract is modeled as a second order digital filter which is a cascade of complex conjugate pole and zero pair networds in the Z-plane (Ref. 2 :175-188). A block

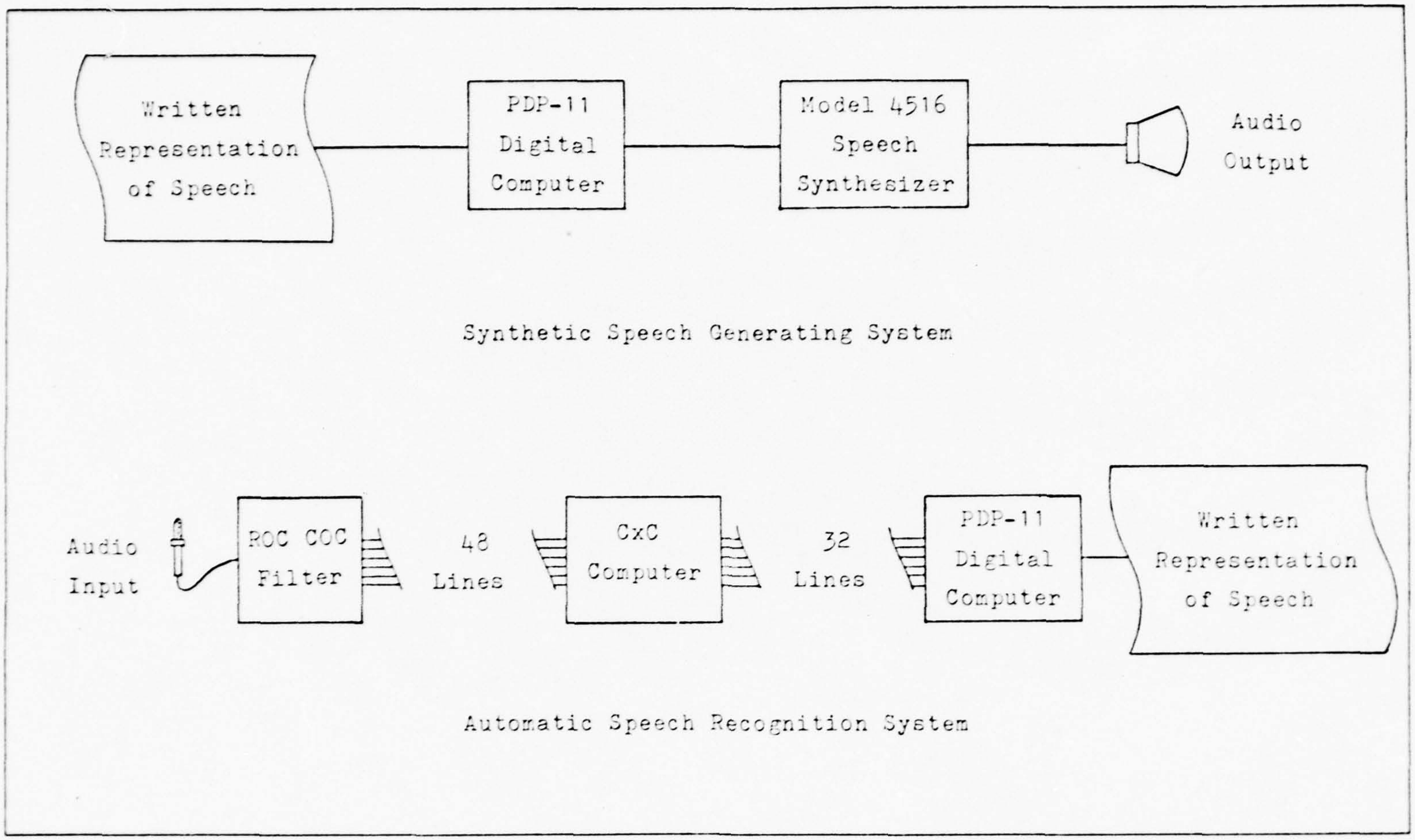


Fig. 2 . Synthetic Speech Recognition System

diagram of the synthesizer is presented in Figure 3 on page 19. The synthesizer requires an input of 24 parameters consisting of frequencies, bandwidths, and amplitudes for each "sound" that it makes. In production of speech these sounds are changing continuously, requiring a new set of parameters every 6-10 ms. The frequencies are converted to locations for poles or zeros on the unit circle in the Z-plane. All pole or zero inputs have an associated bandwidth input. The radial distance inside the unit circle is directly proportional to the value of the bandwidth.

The synthesizer has two normally independent paths. The upper or voiced path includes a pitch impulse generator, a shaping network, a nasal pole and zero network, and a radiation network. This path is used for voiced sounds (vowels, nasals, semi-vowels, and voiced stops), the voiced portion of voiced fricatives, the aspiraat H and whispering. The lower or fricative branch includes a noise generator, a fricative pole and zero network, and a shaping network. This branch is used for voiceless fricatives, voiceless stops, and for the unvoiced portion of voiced fricatives. The two paths are used together for voiced fricatives. The voice path is driven by the noise generator for the aspirant H and whispering. The value of the voiced amplitude (A_V) is the determining factor as to which path is used. A positive A_V triggers a purely voiced or combinational sound. A zero A_V triggers an unvoiced sound. A negative A_V triggers aspiration.

Voice Path. In the production of vowels, nasals, semi-vowels, and voiced stops, the upper path in Figure 3 is used. The value of A_V is set positive and the value of the noise amplitude (A_N) is set to zero.

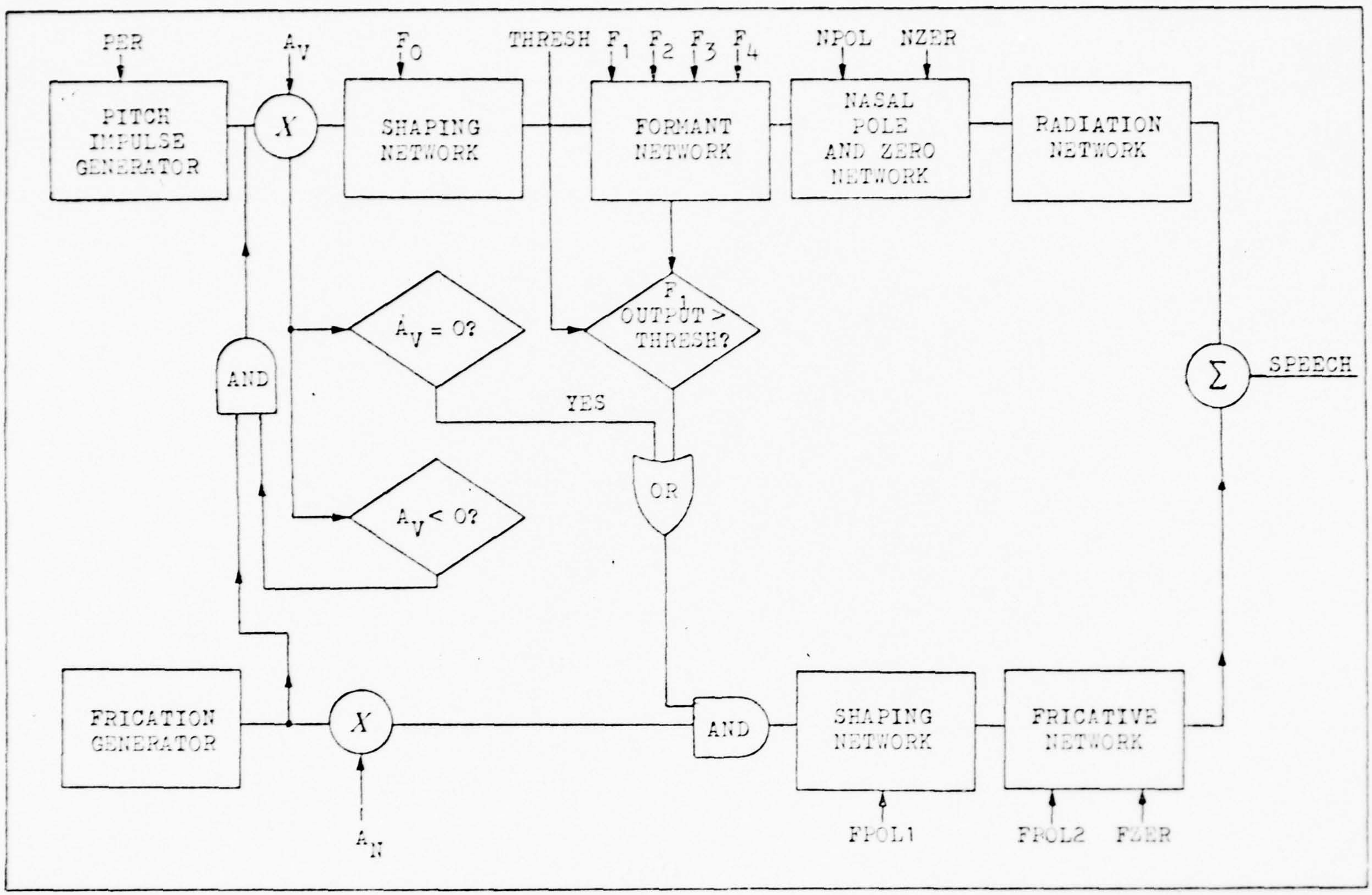


Fig. 3. Block Diagram: Rockland Voice Synthesizer

The amplitude of the voiced output of the synthesizer is directly proportional to the value of A_V.

The pitch period (PER) is in milliseconds and is the inverse of the pitch. This input causes an impulse to be produced at intervals PER apart.

Conceptually the shaping network shapes the impulse into a form resembling the volume velocity waveform produced by motion of the vocal cords and the radiation network simulates the radiation impedance at the lips. However, the shaping network and the radiation network are combined and are represented by two poles on the real axis of the Z-plane.

The three lowest formants (F_1, F_2, F_3) are the crux of the synthesis strategy and are amply explored in Appendix A. The fourth formant (F_4) is set to 3500 Hz and the bandwidths of the four formants are set to 60, 100, 120, and 175 Hz and remain constant for all phonemes except nasals. The bandwidth of F_1 is broadened to 150 Hz for a nasal to simulate the natural dampening of the nasal cavity.

The nasal pole (NPOL) and zero (NZER) are only used for nasals. During a non-nasal sound they are both set to 1400 Hz and effectively cancel each other. Just prior to a nasal NPOL, NZER, and their bandwidths are moved linearly with time to their target values. Just after the nasal they are moved linearly back to the steady state values.

Voiceless Path. The lower or voiceless path in Fig. 3 is used for the production of voiceless fricatives. The output of the frication generator is allowed to pass into the branch by setting A_V to zero and A_N to a positive value. The magnitude of the unvoiced output of the synthesizer is directly proportional to the value of A_N. PER is used to control the duration of the sound. The two poles (FPOL1, FPOL2) and

the zero (FZER) in this branch control the spectral shapes of the noise produced. The "type" of noise produced is an essential characteristic of the fricative being simulated.

Paths in Combination. The two paths are used in combination for the production of voiced fricatives and aspiration. In voiced fricatives, the upper path is excited by impulses and the output of the first formant digital filter is compared to a threshold (THRESH). When the signal is higher than the threshold the output of the noise generator is allowed to pass through the lower path, and the output of the fricative branch is summed with the output of the voiced branch. The net result is similar to a dampened sine wave which has noise added when it is above a certain amplitude. Rabiner's model and hence the Model 4516 are the only synthesizers capable of producing this class of speech sounds.

In aspiration, a negative A_V causes the output for the frication generator to be gated into the voice path. No impulses are produced, and the lower branch is inoperative in this condition. The level of output of the voiced branch is directly proportional to the absolute value of A_V. Aspiration is used in the production of the aspirant H and in the production of whispered voiced sounds.

ROC COC Filter

The ROC COC Filter (short for cochlea) models the sound transformations of both the middle and inner ears. It uses a band-pass filter to simulate the middle ear and a very unique electronic transmission line to simulate the hydro-mechanical functions of the inner ear of the physical system. The middle ear section band-pass filter is

centered at 1500 Hz with 6db/octive skirts (see Fig. 4 on page 23). This filter was designed to fit experimental data (Ref. 9).

According to the designers (Ref. 15:17) the cochlea portion of the ROC COC Filter is ". . . a transmission line with characteristics that vary in a systematic manner along the length of the line. The propagation velocity of a wave traveling along the line changes systematically as a function of distance from the input to termination, becoming ever slower as the wave progresses. In addition, the attenuation characteristic of the line is designed so that high frequencies are attenuated nearest the input while lower frequencies propagate further along the line before attenuation. Both propagation velocity and attenuation frequency vary logarithmically as a function of distance and are related to each other in such a way that a constant number of cycles of a sinusoidal signal are stored in the line between input and the location where the signal is attenuated 40 db. This constant cycle storage is independent of frequency input to the line. By proper manipulation of the design parameters it is possible to design transmission lines with different storage capacities. The storage of a fixed number of cycles in the transmission line, <u>independent of the input frequency</u>, is the feature that distinguishes the class of COC filters from all others. In this type of transmission line we are not trying to obtain the input signal unmodified and delayed in time from various taps along the line. Rather we are interested in observing the modification of the signal that takes place as it propagates along the line." Experimental data (Ref. 9) shows that the physical cochlea stores between 1.5 and 2.0 cycles of the input signal. ROC COC is designed

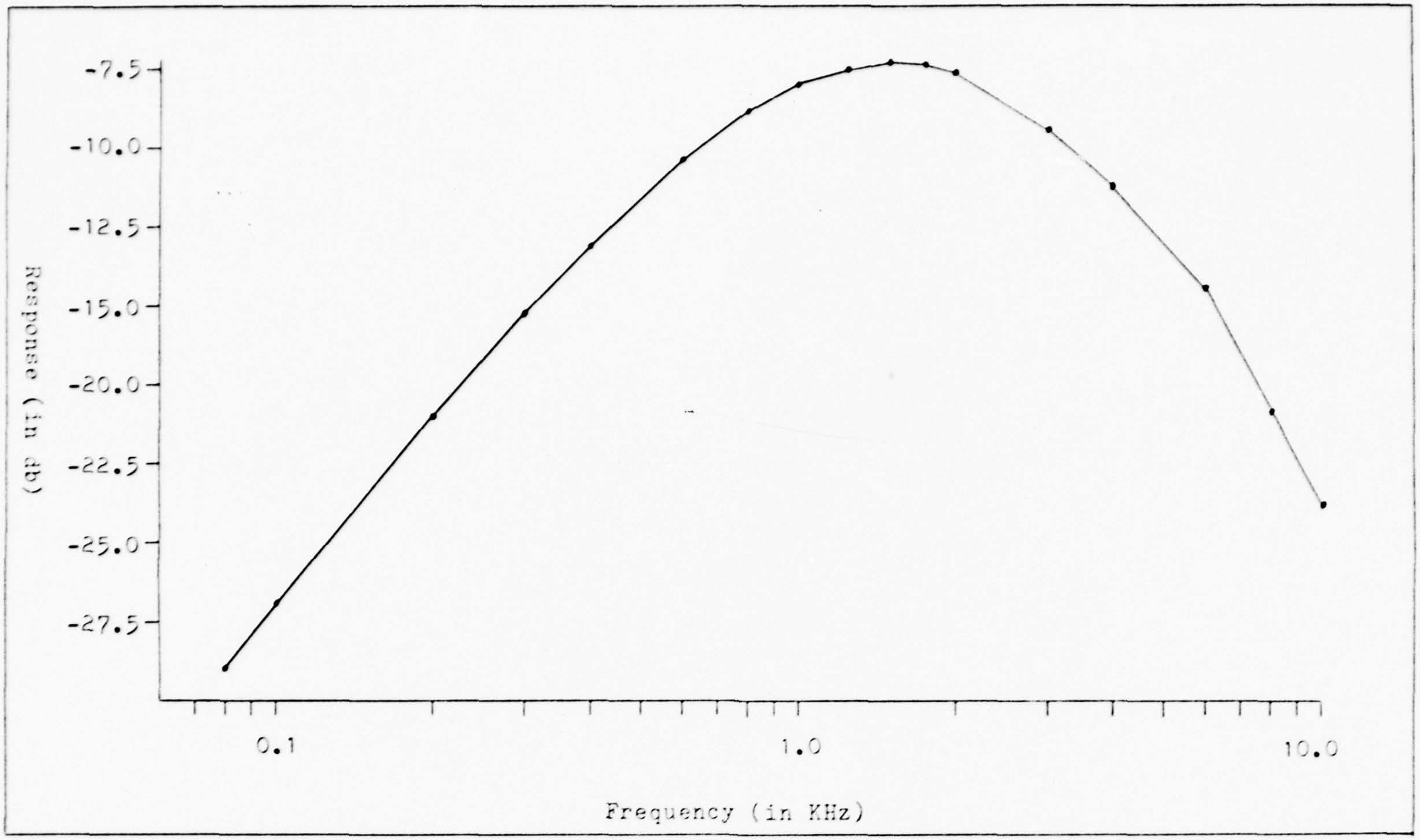


Fig. 4 . Response of Middle Ear Function of ROC COC to 0 db Input

to store 1.75 cycles. Further the ROC COC Filter is designed to have no reflection by having it properly terminated.

The amplitude of the speech signal into the ROC COC Filter must be controlled so that it does not exceed the dynamic range of the instrumentation. This control is done by hand to preserve the amplitude fluctuations in normal speech which are approximately 40 db.

Inside the ROC COC there is a frequency-dependent amplitude envelope in which the signal is contained. Figure 5 on page 25 shows two signals of different frequencies "frozen in time" to demonstrate the amplitude envelope. As the signal passes down the line, it slowly increases to a peak in amplitude. After it passes the peak, it is rapidly attenuated. The position of the amplitude peak in the line is a logarithmic function of frequency with the higher frequencies peaking first and the lower frequencies peaking later.

In the physical cochlea the detector and nerve cells are arrayed along the mechanical line and are so numerous that it can be considered a continuous sampling. Because continuous sampling is impossible to achieve in an electronic model, the ROC COC Filter was designed with 48 taps as a reasonable compromise. It must be kept in mind that it is the modified signals at the various taps that are of interest and the sampling of these signals is the function of the CxC Computer.

CxC Computer

The CxC Computer is a unique piece of hardware that models the logic responses of the nervous system. The computer was designed and developed based on hypotheses that have been experimentally verified but not proven (Ref. 9). CxC is made up of multiples of three basic

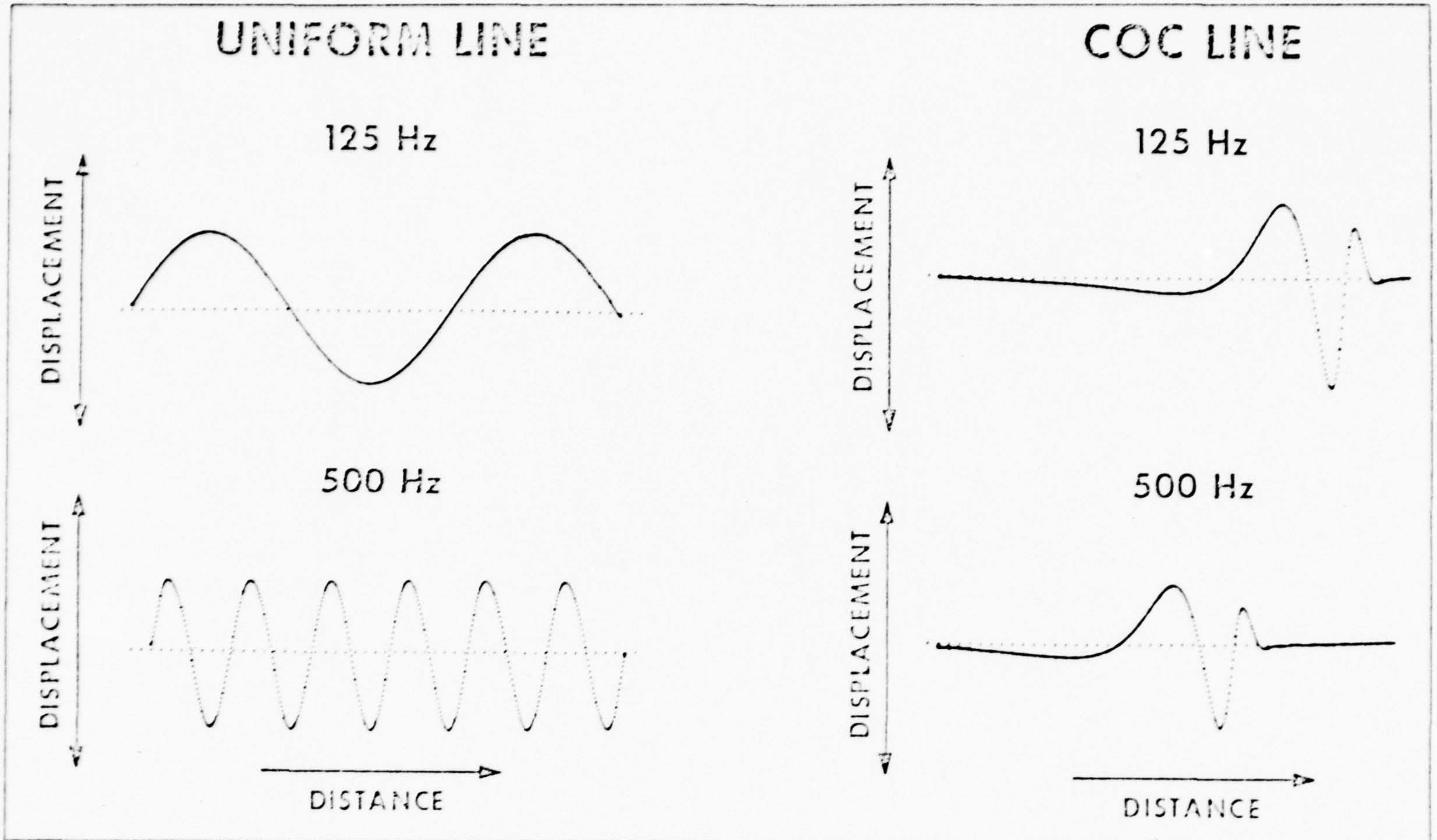


Fig. 5 . Signal Propagation in COC and Uniform Transmission Lines

components hand-wired (or more correctly, hand-patched) together. These components are the syncoders, the synapse buttons, and the sample and hold circuits. Using the three components, a model of an individual neuron or a group of neurons can be achieved. Although not a physical part of CxC, the hardware interface and the PDP-11 digital computer used for data collection are necessary for the operation of CxC.

Syncoders. The basic logic element of the CxC Computer is the syncoder. Functionally the syncoder is a leaky integrator and summing junction followed by a comparator. CxC is a unique type of computer because each syncoder is voltage controllable and its transfer function is signal dependent. The syncoder performs a leaky integration and summation of all inputs. The result is compared to an exponentially decaying threshold and when the two are equal a pulse of given value and duration is generated as an output and the exponential threshold is re-initiated.

Experimentation has shown that this threshold in a real neuron approximates an exponential decay and that the time constant of the decay is a random variable. The model that best fits the experimental data is one in which a new time constant is randomly selected each time the exponential threshold is re-initiated. Once a time constant is selected, it is not changed until the threshold is again re-initiated. However, producing CxC with such random syncoder elements was not technically nor economically feasible. Therefore, AMRL designers decided to make the syncoders deterministic by fixing the time constant of each given unit. However, different syncoders may be set with different time constants.

Once the neuron (syncoder) produces a pulse it goes into a "positive refactory period" for the duration of the output pulse length. That is, the threshold goes to infinity and the neuron (syncoder) cannot be fired regardless of the input. Thus, the response to a DC level input is a periodic string of pulses. The response to a time varying signal is complex and depends upon integration time, threshold decay constant, and refractory time of the syncoder, all of which are adjustable on each syncoder. These parameters are adjusted according to the use of the particular syncoder in the network. The syncoders that are used as detectors on the 48 taps of the ROC COC Filter are set so that they will fire on each peak of the highest frequency that can reach that particular tap at maximum input amplitude. Because the syncoders continuously compare input to the threshold they are obviously amplitude dependent.

Synapse Buttons. A synapse button is connected to the pulse output port of a syncoder. It is basically a switch that conducts when the pulse output of the syncoder is high. The outputs of these switches are normally connected to the integrating inputs of other syncoders. Therefore, when a syncoder fires, the switch conducts and a voltage applied to one side of the switch appears at the integrator input of syncoder. A voltage is produced at the syncoder summing junction and the voltage exponentially increases while the switch is closed and immediately begins an exponential decay when the switch opens. Pulses can easily be weighted or assigned relative significance by controlling the voltages to the synapse buttons. Each pulse output can "fan out" to eight inputs.

Sample and Hold. The sample and hold (s&h) circuits supply the voltage sources required by the synapse buttons and DC levels that are added at the summing junctions to bias the various time-varying signals. These circuits are controlled by a PDP-8/S digital computer. The PDP-8/S addresses each s&h board individually and supplies a predetermined voltage to that s&h board through a digital to analog converter. The PDP-8/S requires less than two minutes to sequentially address all 1728 s&h boards in CxC. The voltage on a s&h board after the required two minutes is about 97% of the original voltage.

Hardware Interface. The Asyncronous Pulse Pattern Processor (ASPPP) is the hardware interface used to sample up to 32 pulse outputs from CxC and store the results in a PDP-11/20 digital computer. Each five microseconds the ASPPP looks for up to two rising edges of pulses on the 32 channels starting with the first channel output of CxC. If it finds at least one, it records the channel(s) and the time since the last pulse was recorded on any channel. Although the ASPPP can only record the first two pulses (in channel order) in a five microsecond time section, this has proven to not be a cause of significant loss of data. In an informal inspection of speech data it was found that two channels had fired "simultaneously" less than 5% of the time. Therefore, the amount of data lost due to a third simultaneous firing must be extremely small.

Programs. The ASPPP presents the data received from CxC to the PDP-11 computer but computer programs are needed to accept the data and control the sampling. AMRL has several such programs, one of which is used extensively in this project. This program starts data collection when a switch is manually depressed and stops data collection when the

switch is released. Data collected by this program is continuously recorded on a disk. Other available programs can display the data on a CRT and can statistically analyze the data in several ways. Although these latter programs were exceptionally useful in the basic research for this project, they are not used in the final product.

Pitch Period Marker. The first channel of CxC output is devoted to a pitch period marker. A pulse on this channel indicates that a voiced sound is present and the approximate beginning of each pitch period. The input signal is low-pass filtered to 300 Hz. A voltage proportional to the average amplitude is generated by full wave rectification and integration of the filtered signal and this voltage is used to bias a syncoder which receives the filtered signal as an input. This syncoder responds to the large amplitude peaks that occur at the beginning of each impulse excitation.

Amplitude Channel. The second channel of CxC output is devoted to a measure of input signal amplitude. Pulses on this channel are generated by CxC at a rate logarithmically proportional to the amplitude of the input signal. A short-interval average of the signal is placed on an input to a syncoder. The syncoder by its very nature will respond at a rate logarithmically proportional to the level of a DC input.

Sensory Syncoders. The syncoders that are used as detectors on the 48 taps of the ROC COC Filter are adjusted (integration time, threshold decay constant, refractory time, biasing, and feedback) so that they will fire on each peak of the highest frequency that can reach that particular tap at maximum input amplitude. The response of these syncoders to a periodic signal is always periodic but the number

of pulses in a sequence varies. The number of pulses in a period can vary from one (evenly spaced pulses) to at least seven.

CxC Output

This speech recognition process operates on the pulse patterns generated by the networks of CxC. Therefore, at least a conceptual understanding of this data and what it "looks" like are critical to understanding this thesis. Figures 6 through 11 are examples of the CxC responses to sinusoids and square waves of 500, 1000, and 1500 Hz. They are plots of the pulses on the 32 pulse output channels of CxC versus time. The location of the pulses in time is an indication of when the pulses occurred. In these figures the high frequency components of the sound are in the lower portion of each plot and the lower frequency components are seen at the top. A square wave has high frequency components at the rising and falling edges and lower frequency components throughout. Thus, in the plots of the square waves a long chain of pulses is apparent at the leading edge and a short string of high frequency component pulses are apparent at the falling edge. One can also note that the lower frequencies propagate further down the ROC COC and also did not begin firing channels as soon as the higher frequencies did. In both the square and sine wave figures, the changing velocity of the waves is also readily apparent as a curvature in the pulse patterns. If the velocities of the signals had remained constant the pulse patterns would have approximated a sloped, straight line.

Figures 12 through 23 on pages 38 through 49 present the outputs of CxC for several natural and synthetic speech sounds. Once again, the

high frequency components of the sounds are seen in the lower portion of each plot and the lower frequency components are seen at the top. It can be seen that the synthetic patterns compare with their natural counterparts. It can also be seen that the different sounds produce a wide range of pulse patterns. From this wide range of pulse patterns it was believed that there must be a way of recognizing what sounds were being made either on a pitch period basis (for voiced sounds) or on a small sample basis (for unvoiced sounds). This recognition is the subject of Chapters III and IV.

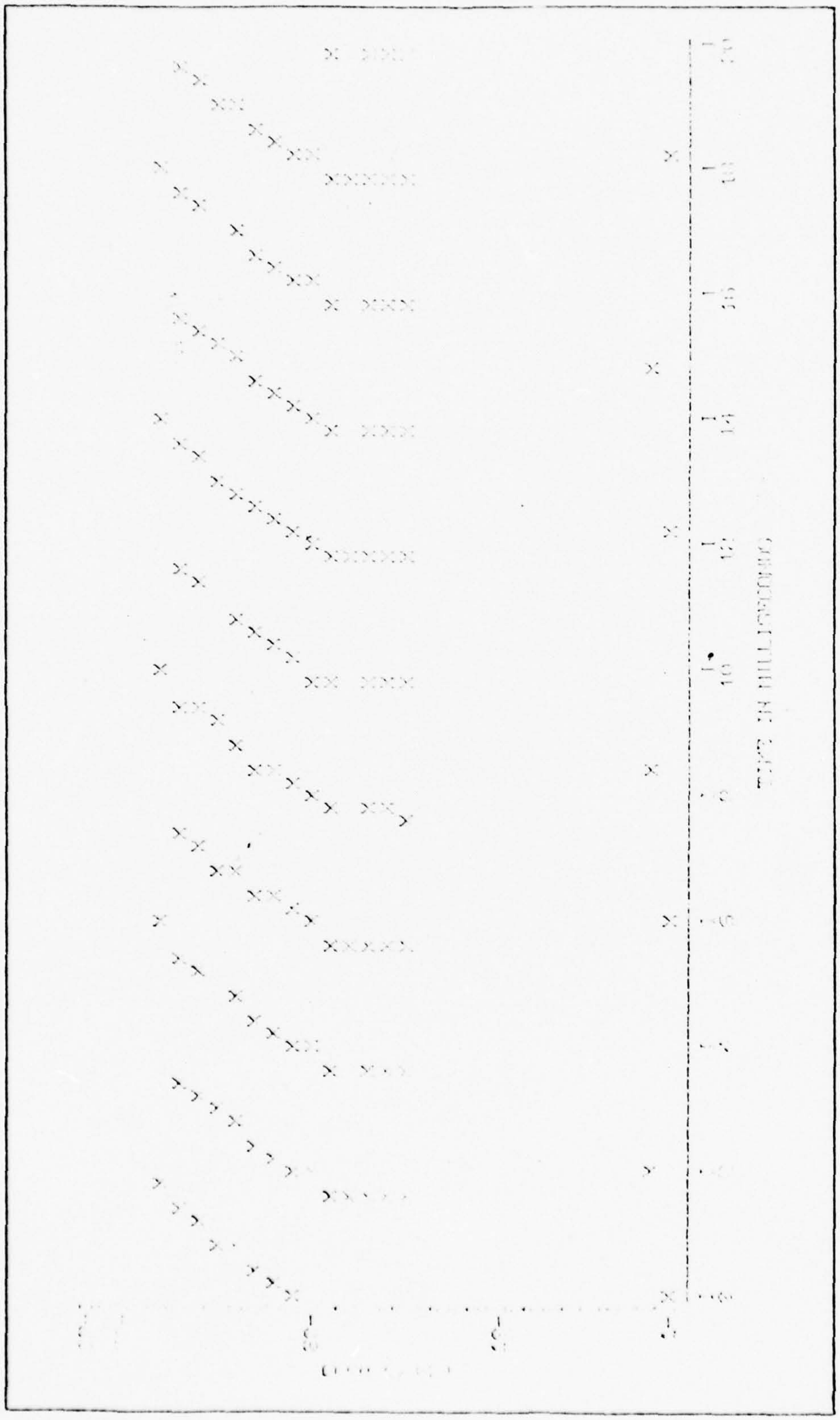


Fig. 6 . CxC Output for 500 Hz Sine Wave

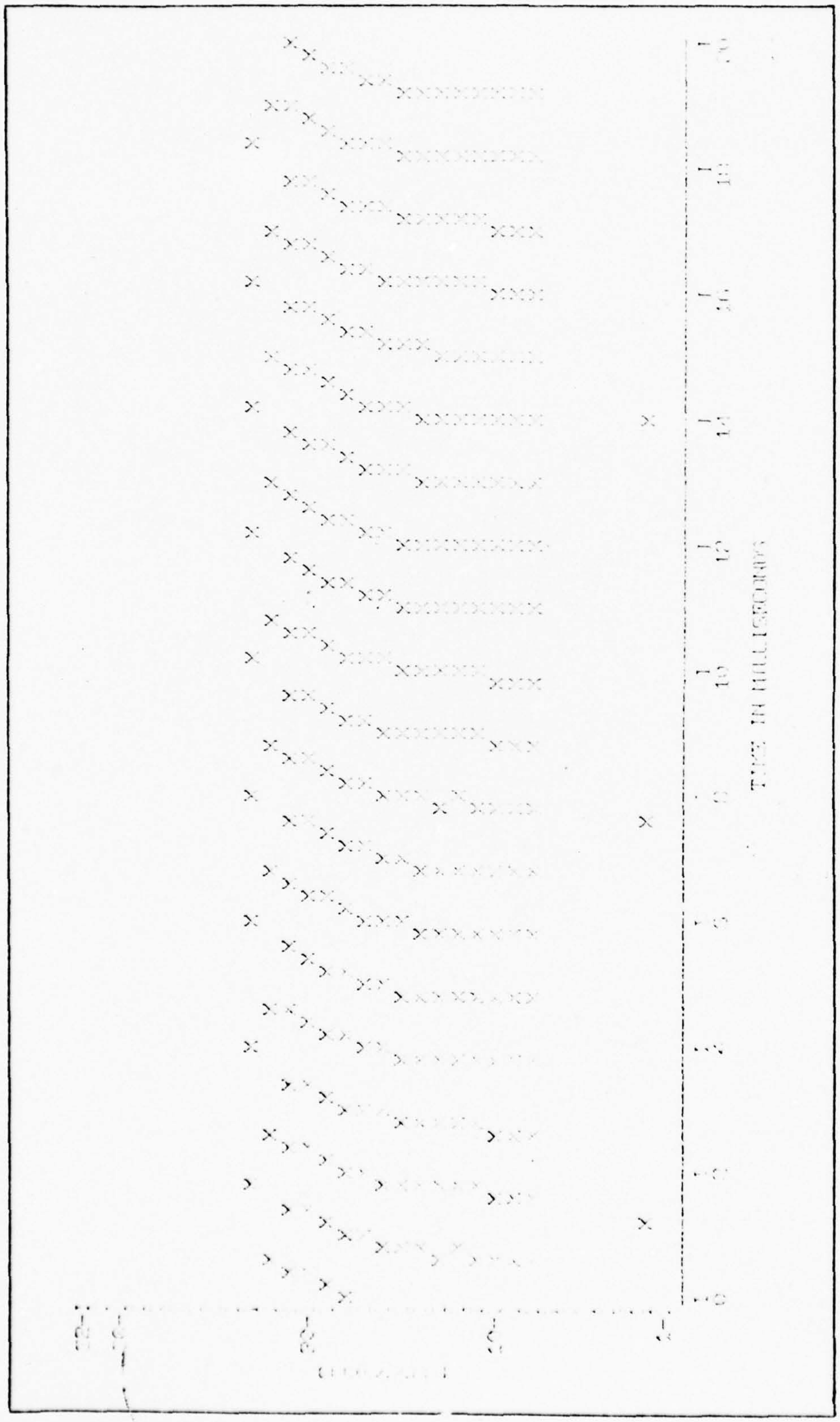


Fig. 7 . CxC Pulse Output for 1000 Hz Sine Wave

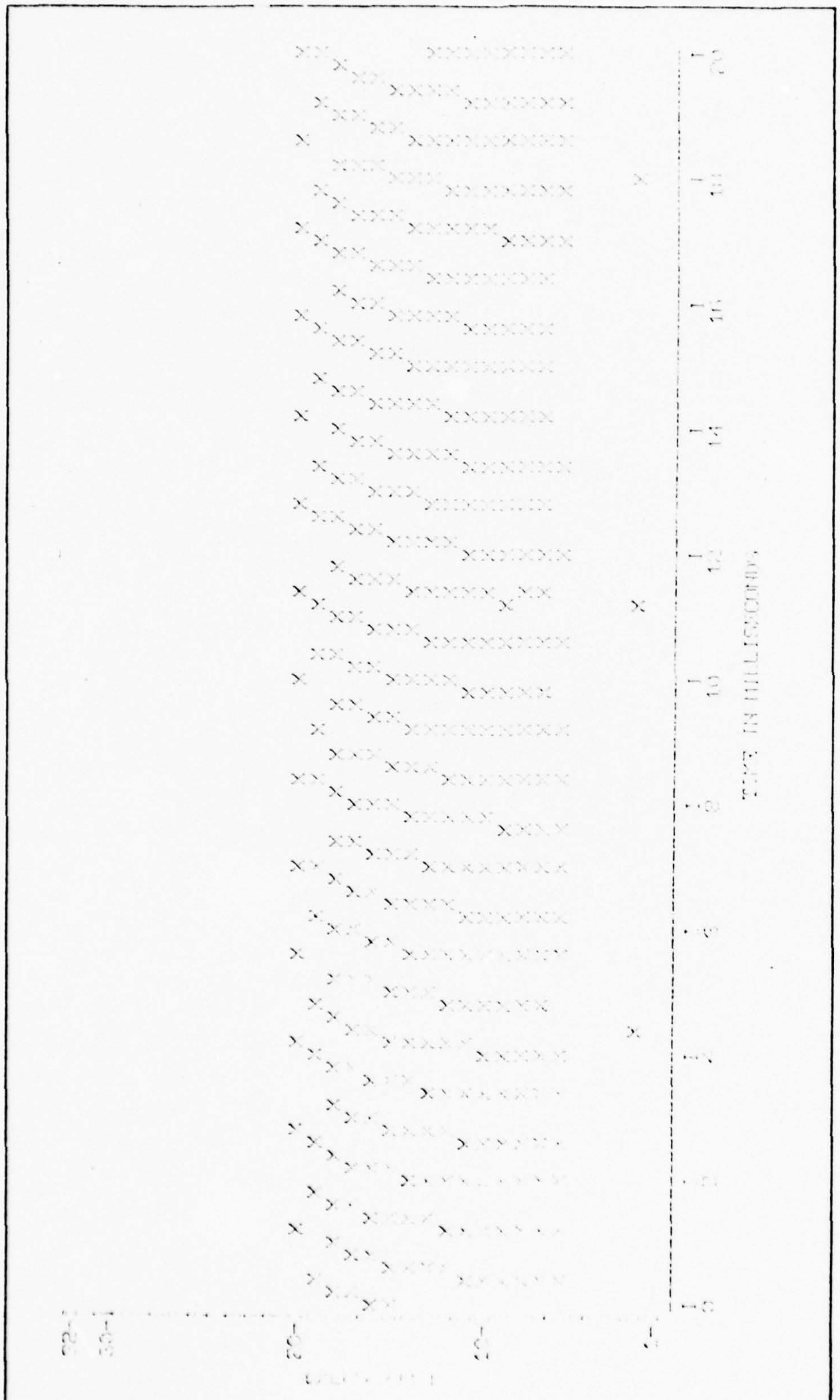


Fig. 8 . CxC Pulse Output for 1500 Hz Sine Wave

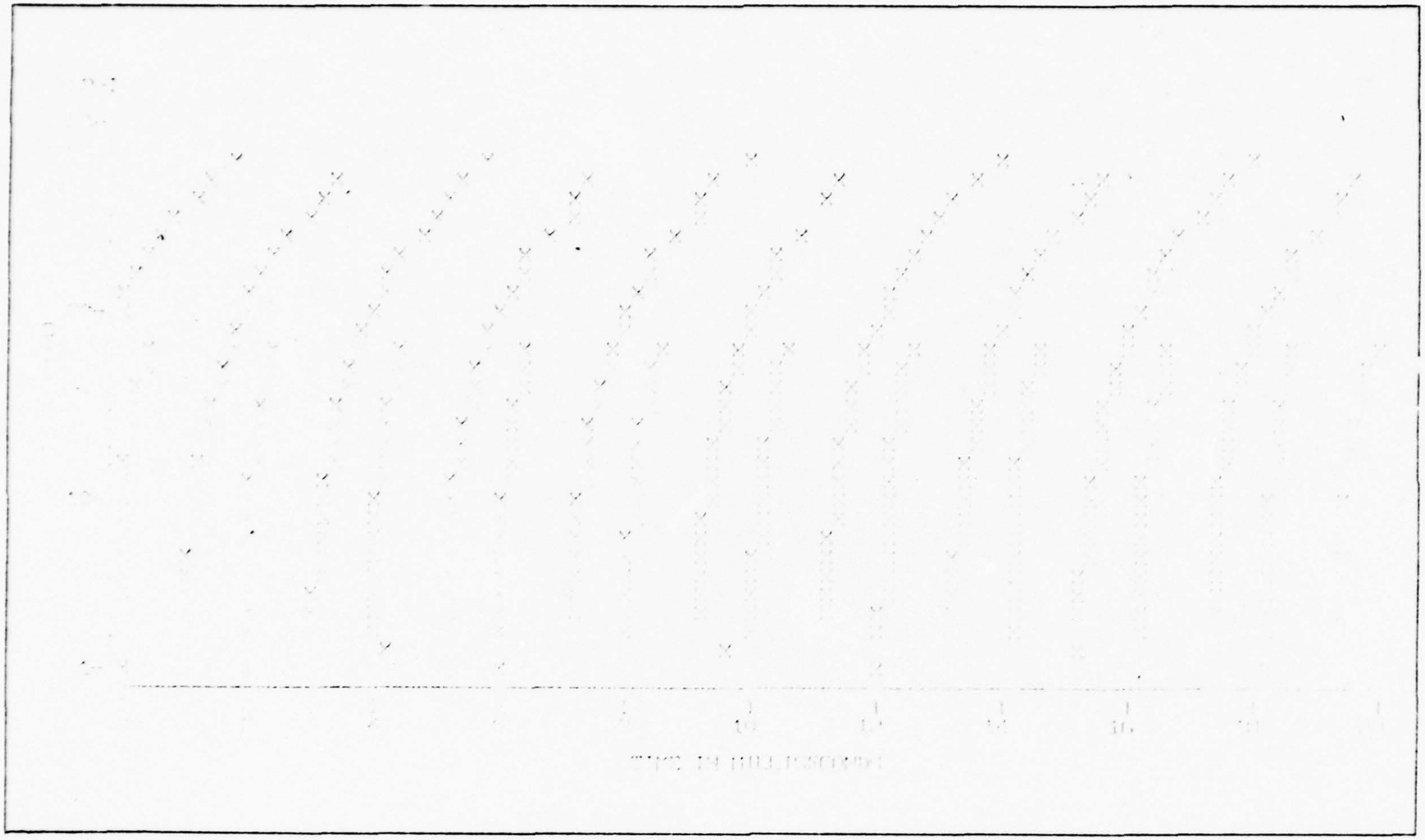


Fig. 9 . CxC Pulse Output for 500 Hz Square Wave

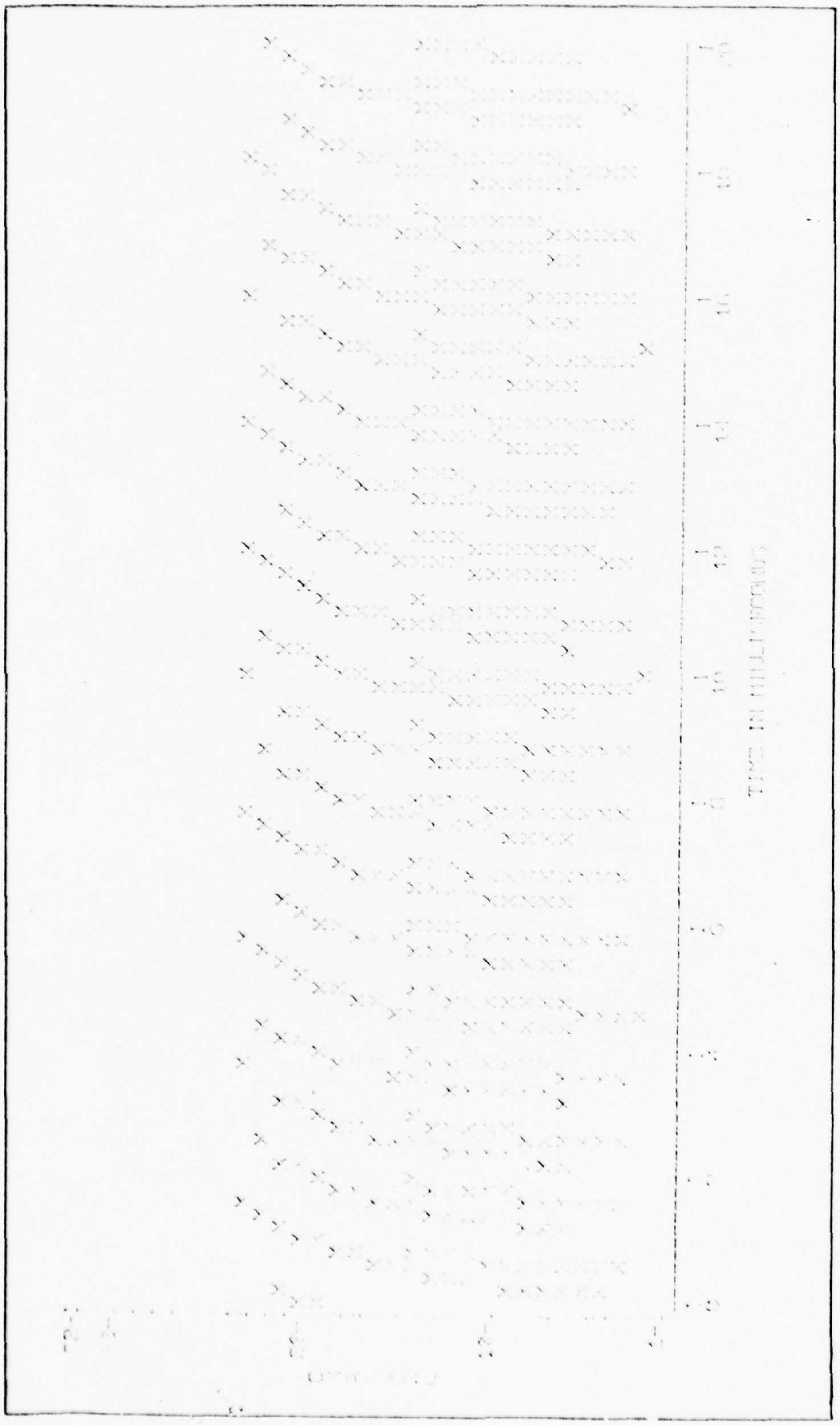


Fig. 10. CxC Pulse Output for 1000 Hz Square Wave

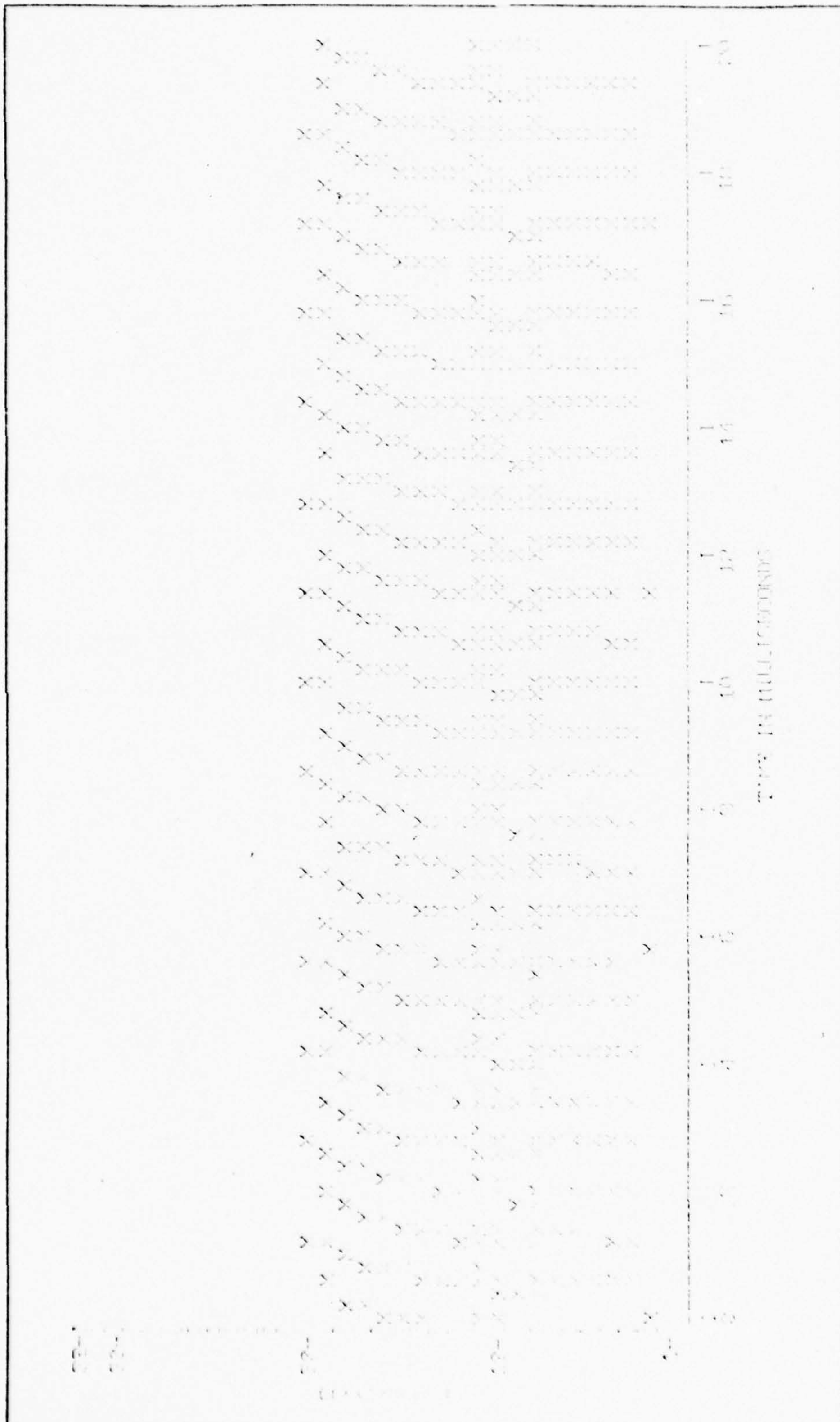


Fig. 11. CxC Pulse Output for 1500 Hz Square Wave

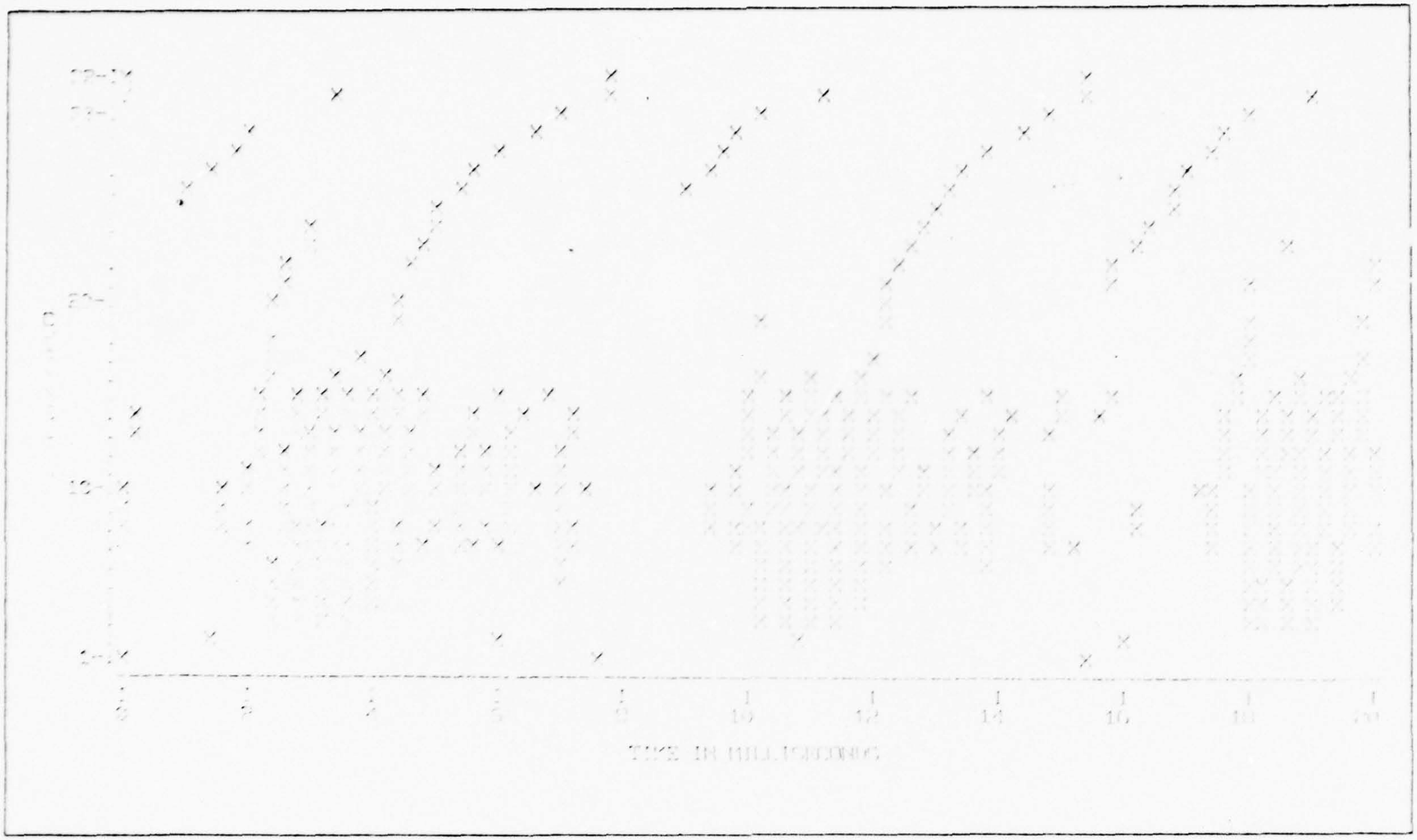


Fig. 12 . CxC Pulse Output for Natural IY

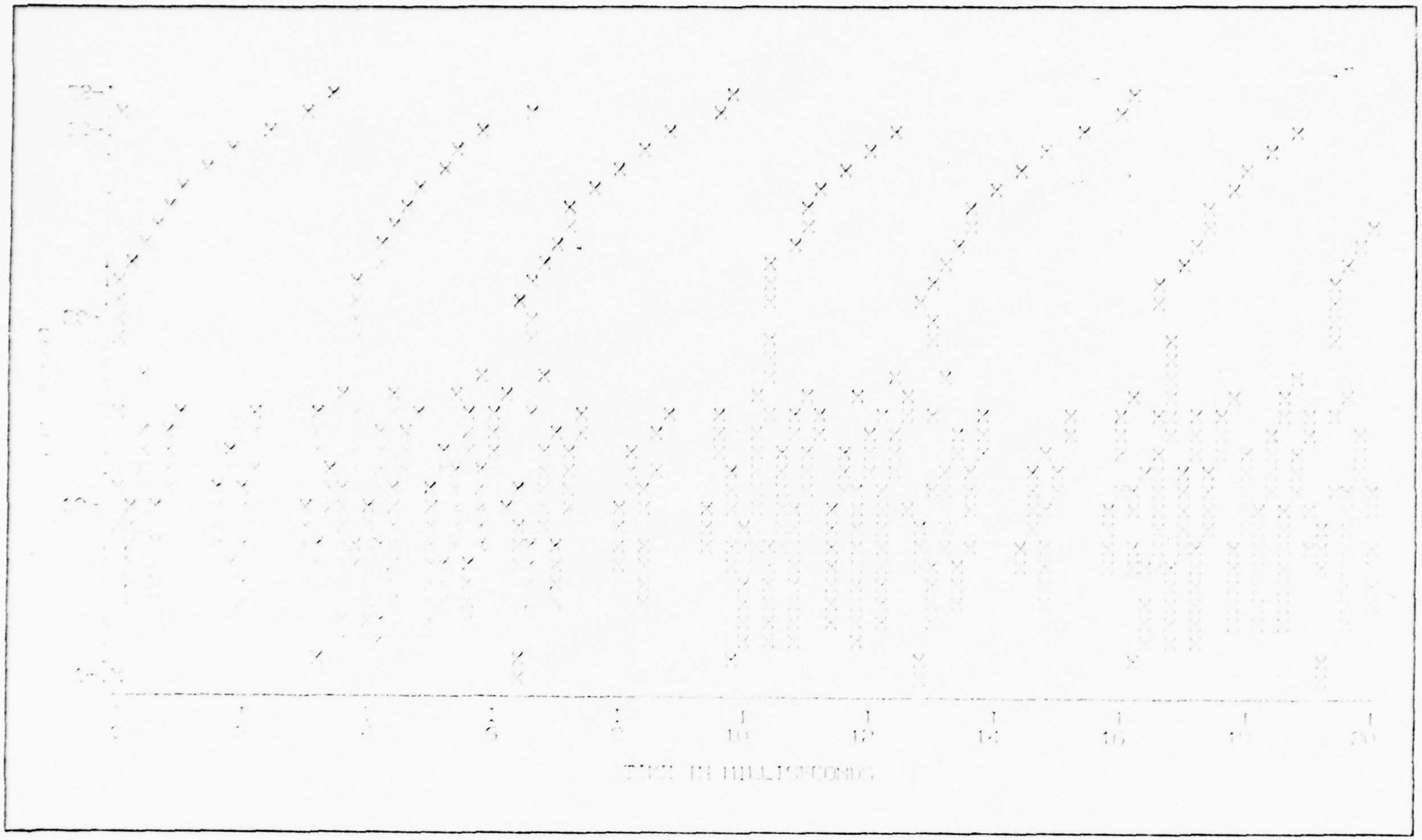


Fig. 13 . CxC Pulse Output for Synthetic IY

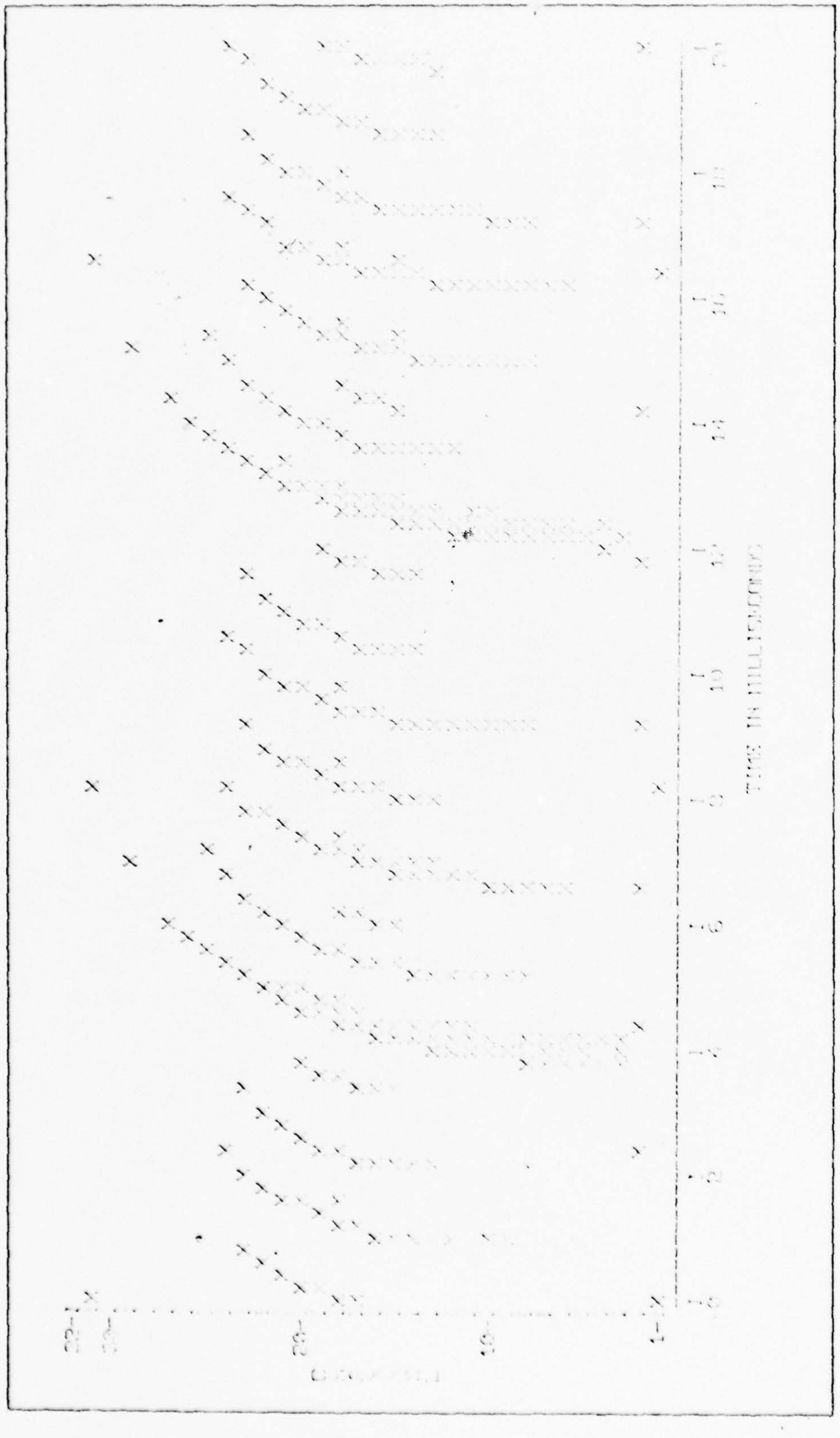


Fig. 14. CxC Pulse Output for Natural AA

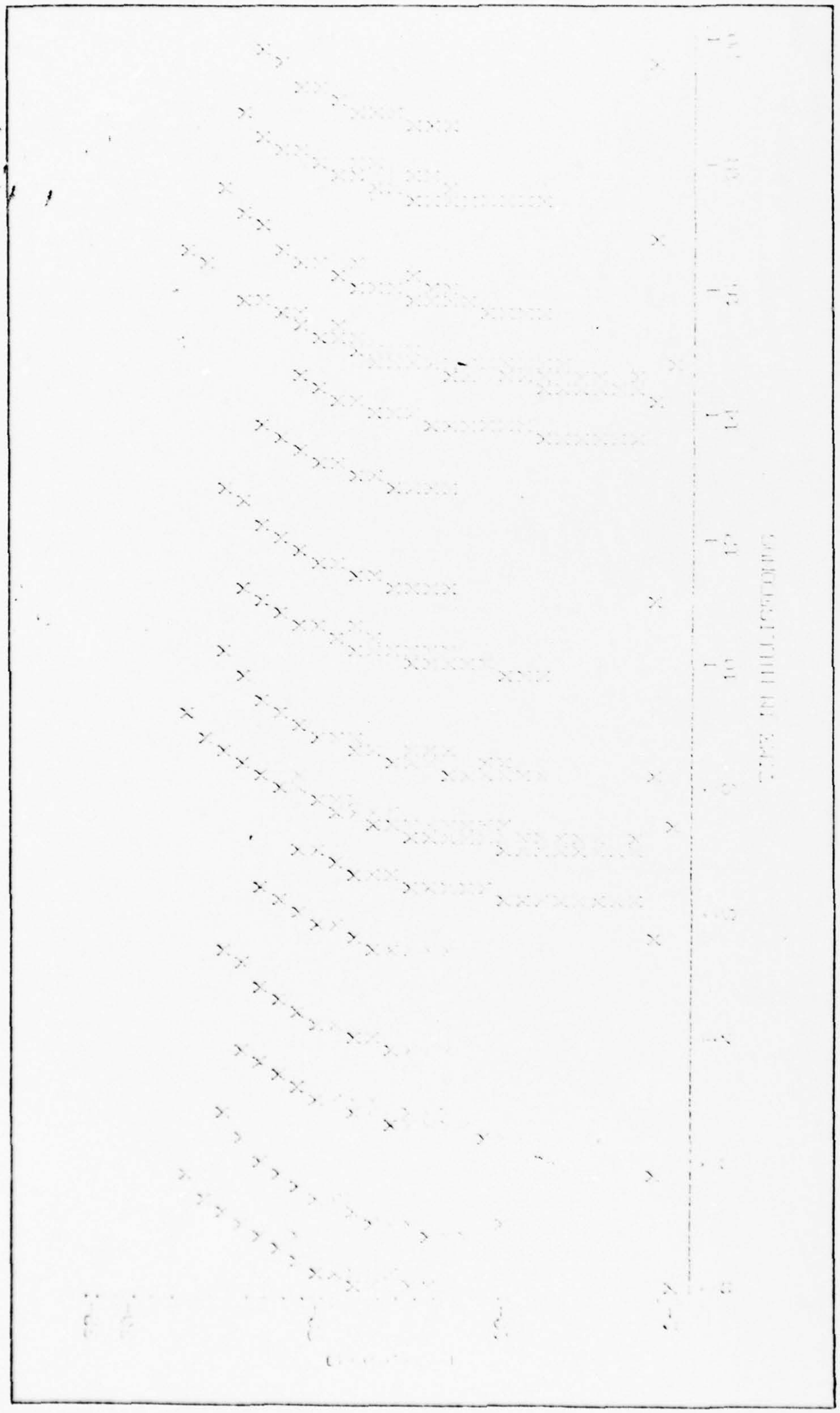


Fig. 15. CxC Pulse Output for Synthetic AA

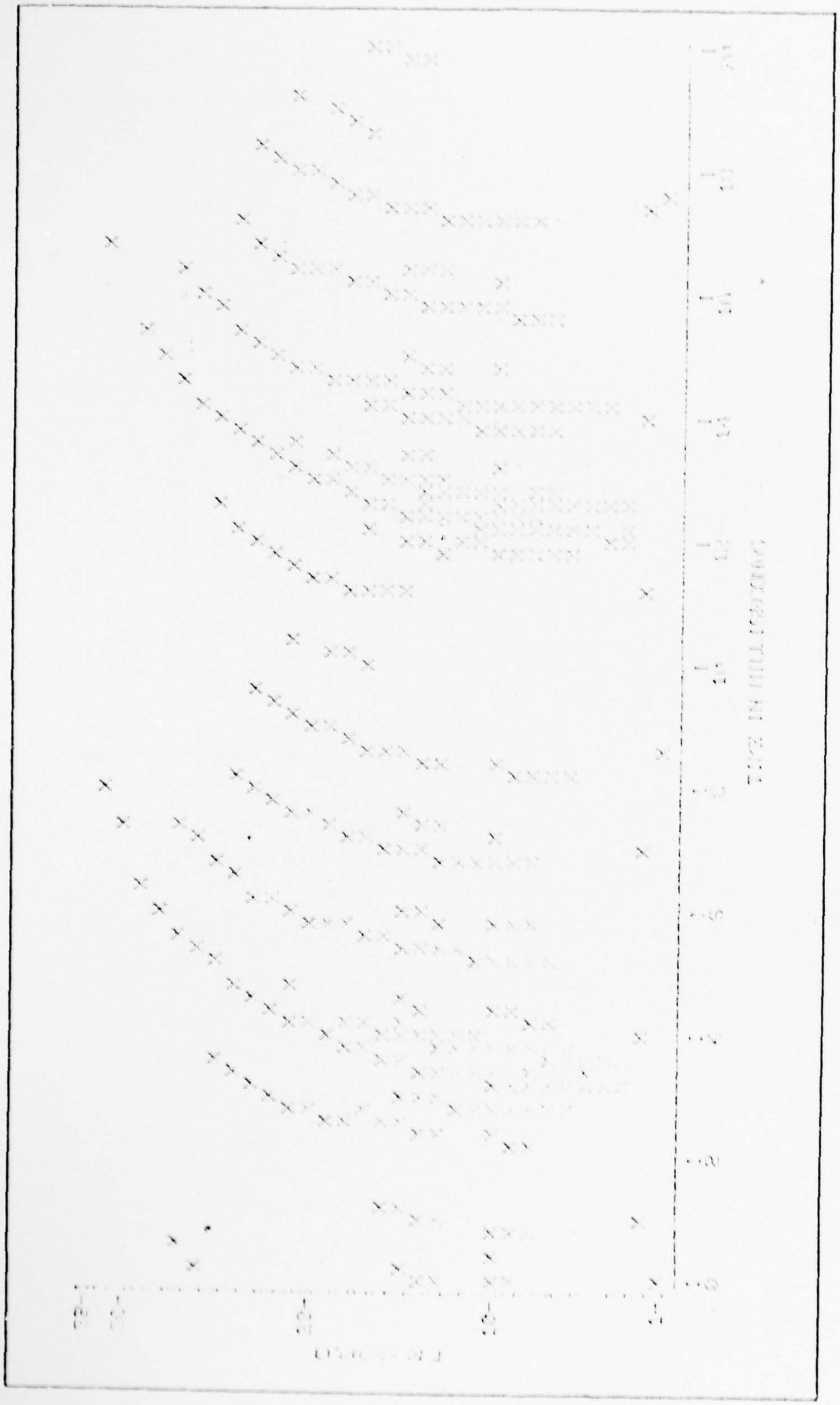


Fig. 16. CxC Pulse Output for Natural AE

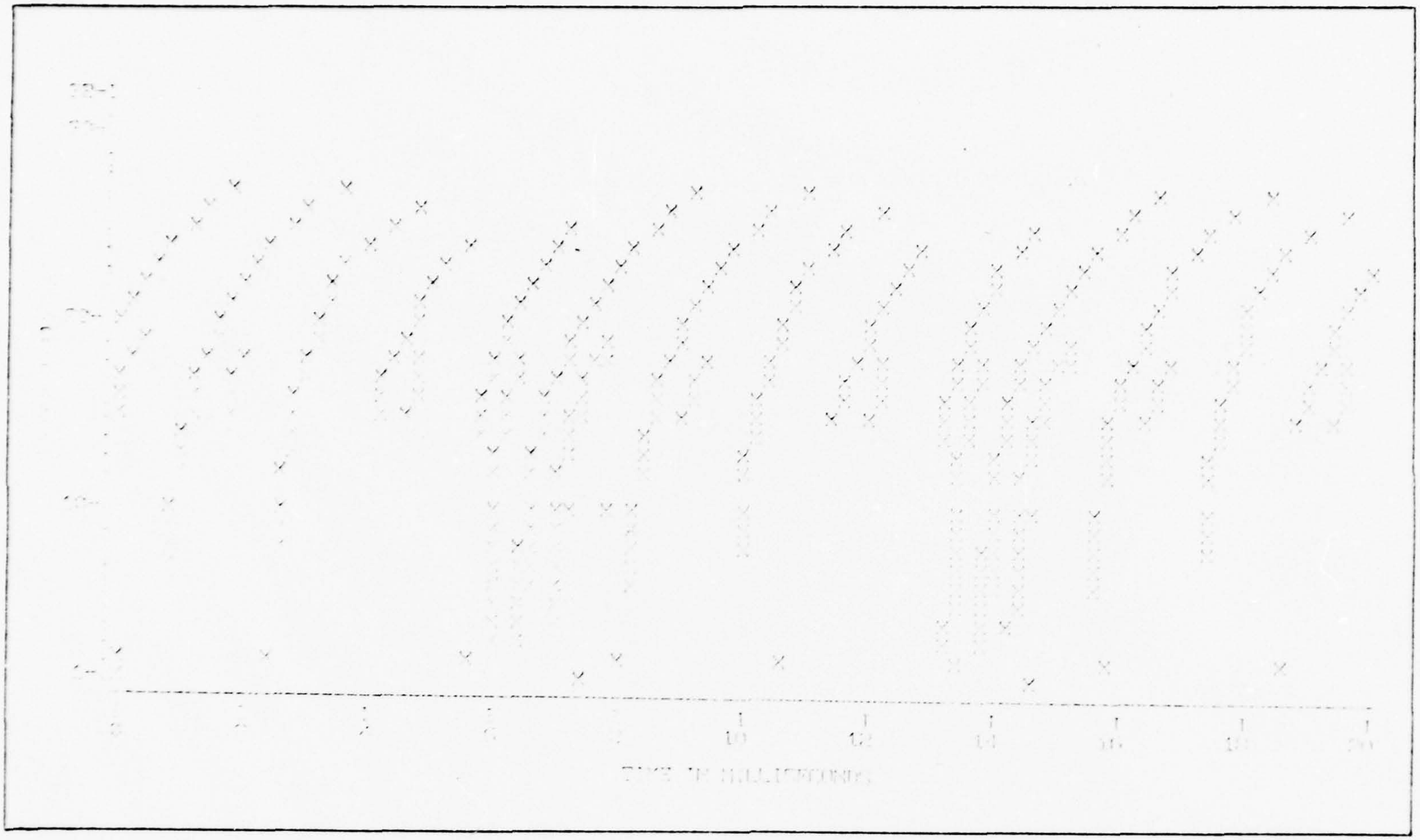


Fig. 17. CxC Pulse Output for Synthetic AE

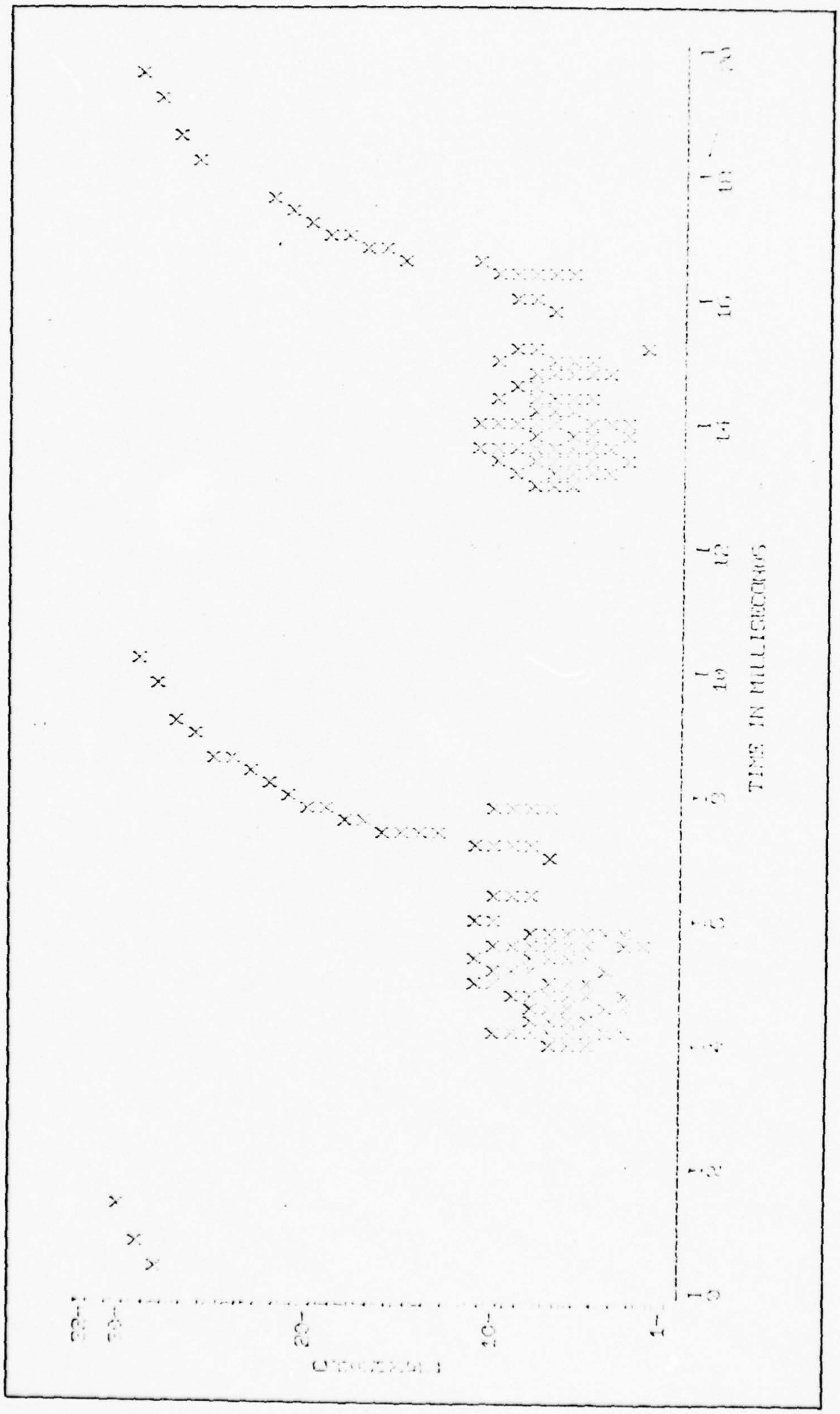


Fig. 18 . CxC Pulse Output for Natural 2Z

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

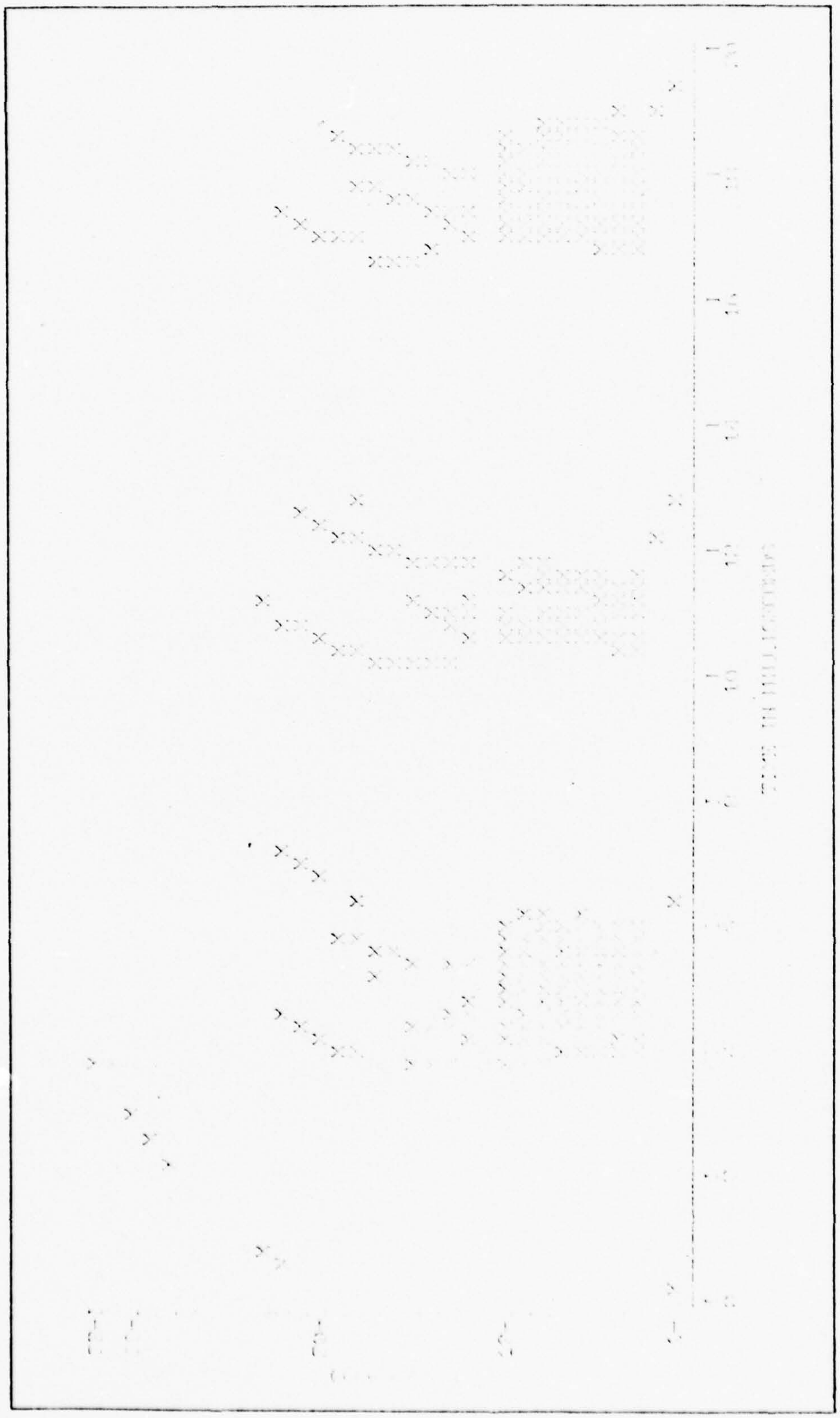


Fig. 19. CxC Pulse Output for Synthetic ZZ

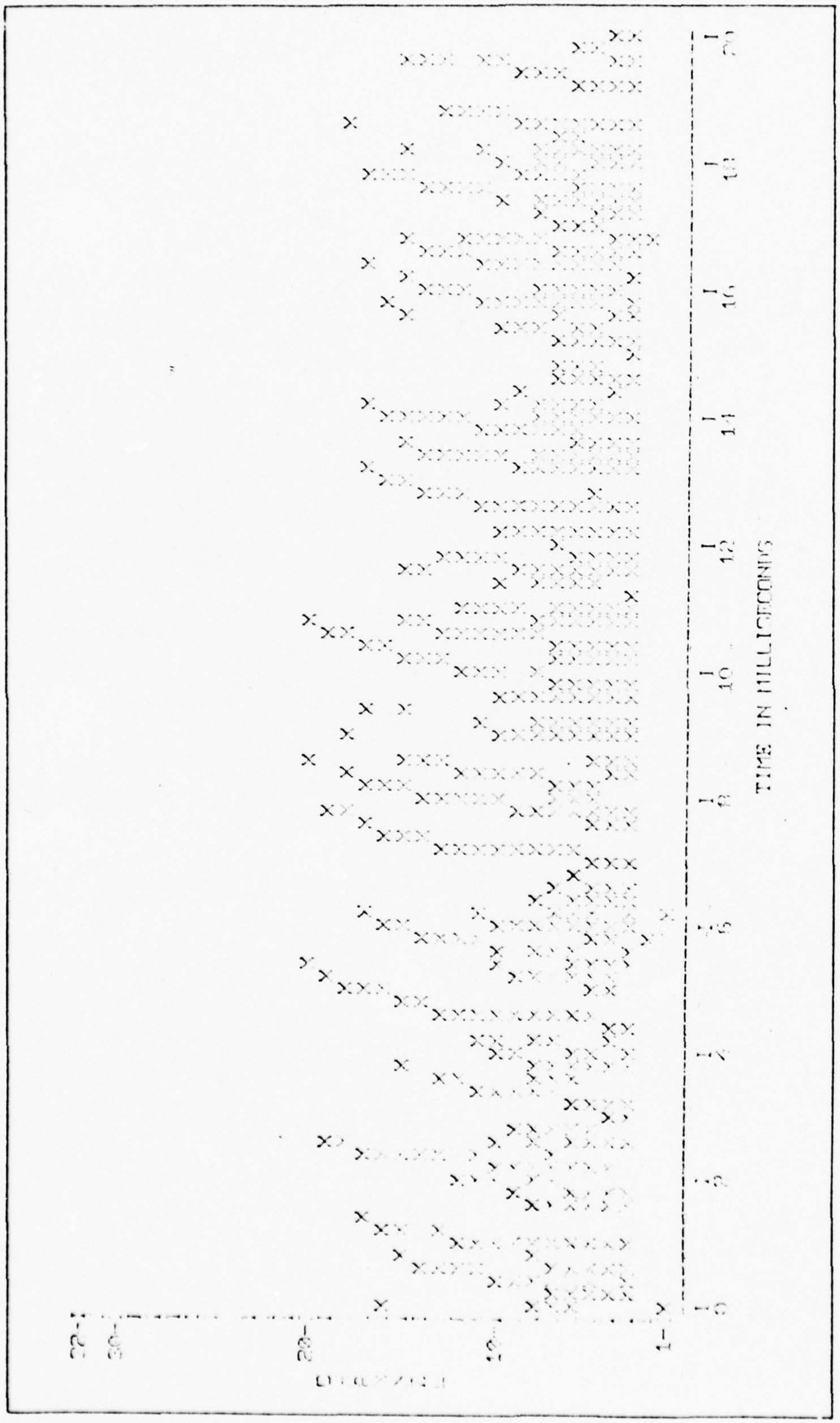


Fig. 20. CxC Pulse Output for Natural FF

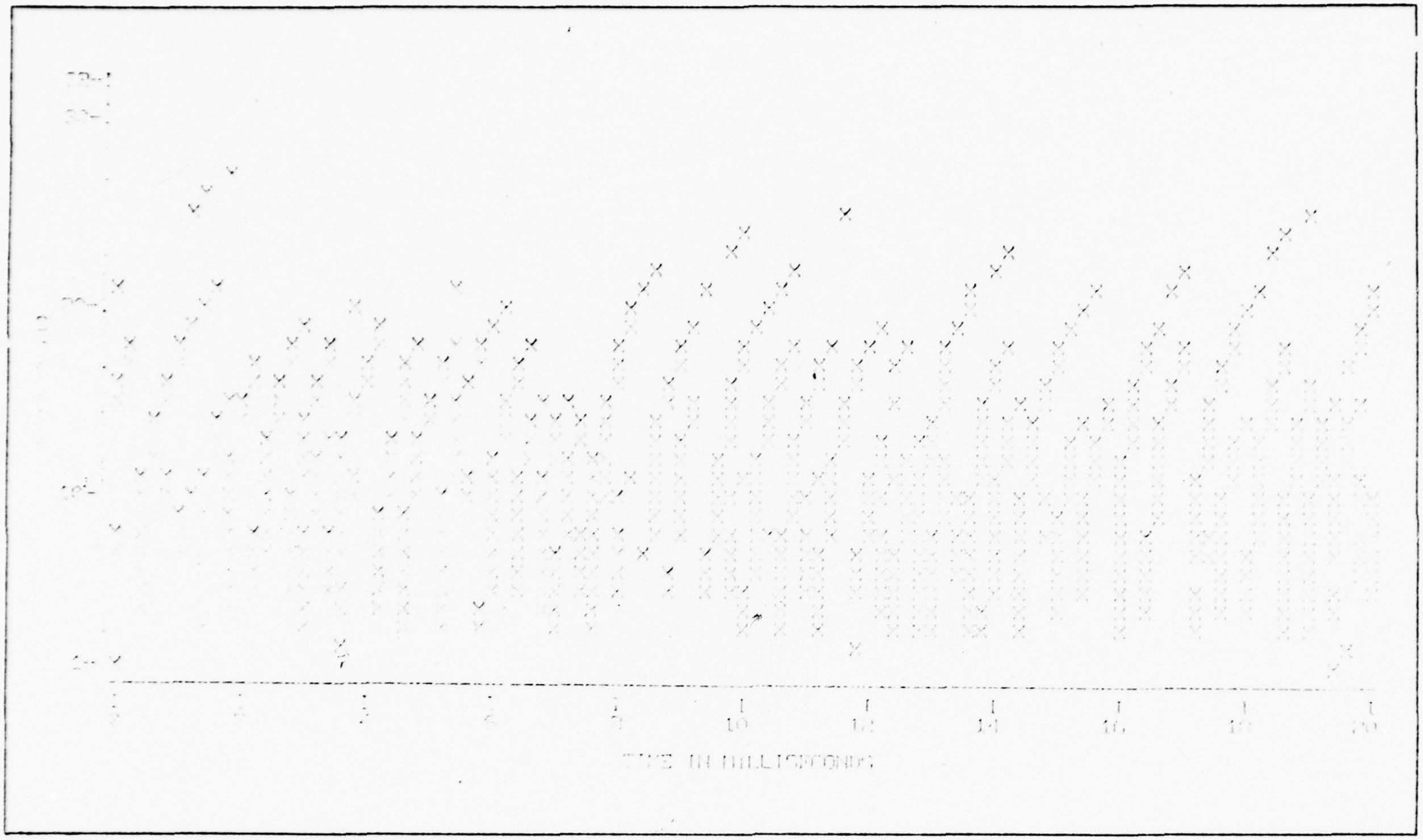


Fig. 21 . CxC Pulse Output for Synthetic FF

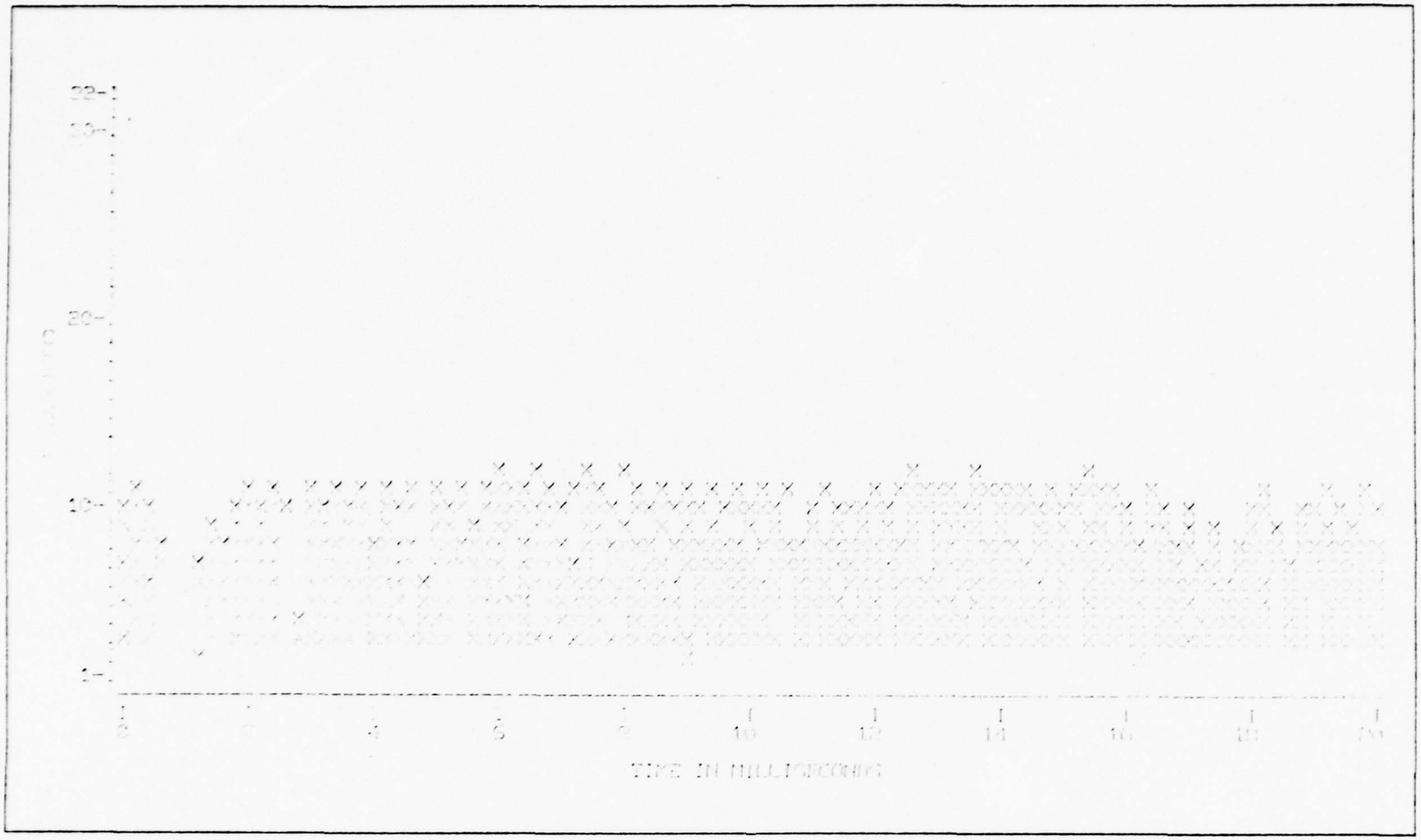


Fig. 22 . CxC Pulse Output for Natural SS

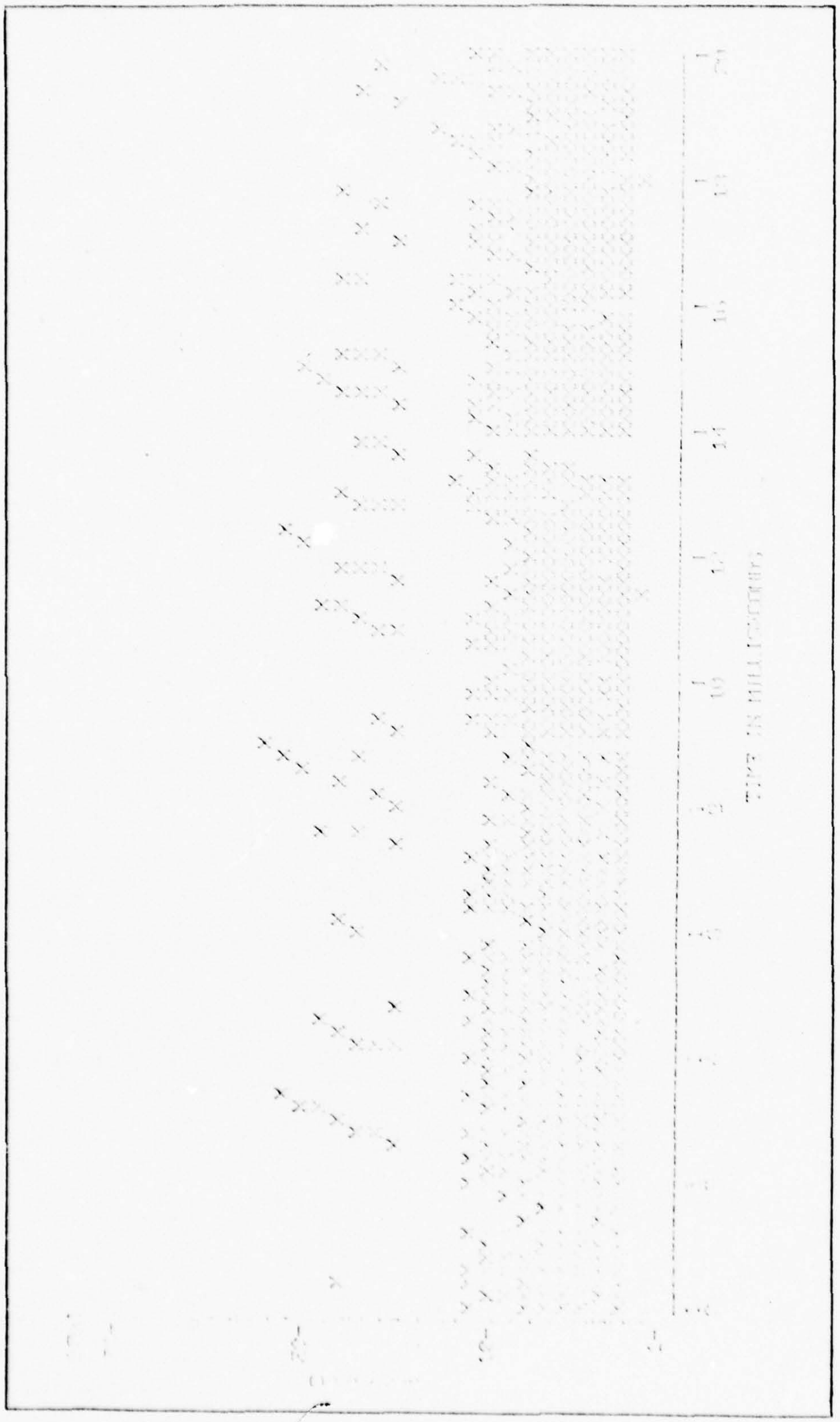


Fig. 23. CxC Pulse Output for Synthetic SS

III. Segment Identification

In this chapter the methods of recognizing small segments of synthesized speech from the data collected from CxC are presented. The segments for voiced speech are delineated by pitch-period marker pulses. For a male speaker or for the output of the speech synthesizer used here, these pulses are from six to ten milliseconds apart. The pitch period marker allows the natural periodicity of voiced speech to be used for segmentation. When the pitch-period marker pulses are absent, indicating voiceless speech, data is analyzed in ten millisecond time segments. Silent periods, no pulses on any channel, are analyzed as a single unit regardless of their length. Analysis of each segment is based on the time between pulses on each channel and the number of times each channel fired.

Initial Manipulations

After data from CxC is retrieved from disk storage, two initial manipulations are performed on each segment. The first of these manipulations is a pulse interval determination and the second is a channel firing statistic.

Pulse Interval Determination. There are 30 channels of data output from CxC. The data on each channel are pulses of constant amplitude and duration produced by a syncoder operating in a specific network. The ASPPP records the time of the rising edge of a pulse and the channel on which it occurred. The most obvious data manipulation is to determine the time between pulses on a particular channel. The duration of this "pulse interval" is limited to between 0.01 ms (100,000 Hz) and 4.80 ms (208 Hz). These limits were chosen based on known speech frequencies

and on an analysis of the range of intervals in the data from CxC. Each pulse interval considered is rounded off to the next lower 0.01 ms increment and recorded in a linear matrix of 480 elements. Pulse pattern determination is done without regard to the channel on which the pulses occurred. For each segment a histogram of the number of occurrences of each pulse interval is generated. The histogram is normalized so that there are a total of 300 pulse interval occurrences in each histogram. Typical histograms for a single pitch period of IY and AA and a ten millisecond segment of SS are displayed in Figures 24 through 26 on pages 52 through 54 .

Channel Firing Statistic. The second initial process is to determine how many times each of the 30 data channels fired (produced pulses) within the current segment. The result is stored in a linear matrix of 30 elements.

Speech Categorization

For convenience, speech is divided into three categories and two special cases. The first category is steady-state speech; that is, sounds which occur at or very near steady-state values for either several pitch periods (voiced sounds) or for an extended length of time (voiceless sounds). The second category is dynamic speech, which is characterized by rapid changes in the speech patterns. The six stops (B, D, G, P, T, and K) are the sole members of this division. The third category is the aspirant H which is a unique sound in American speech. The two special cases are a stop in utterance initial position and a stop in utterance final position. These are special because the initial "shut down" portion of the stop will be missing when the stop is in the

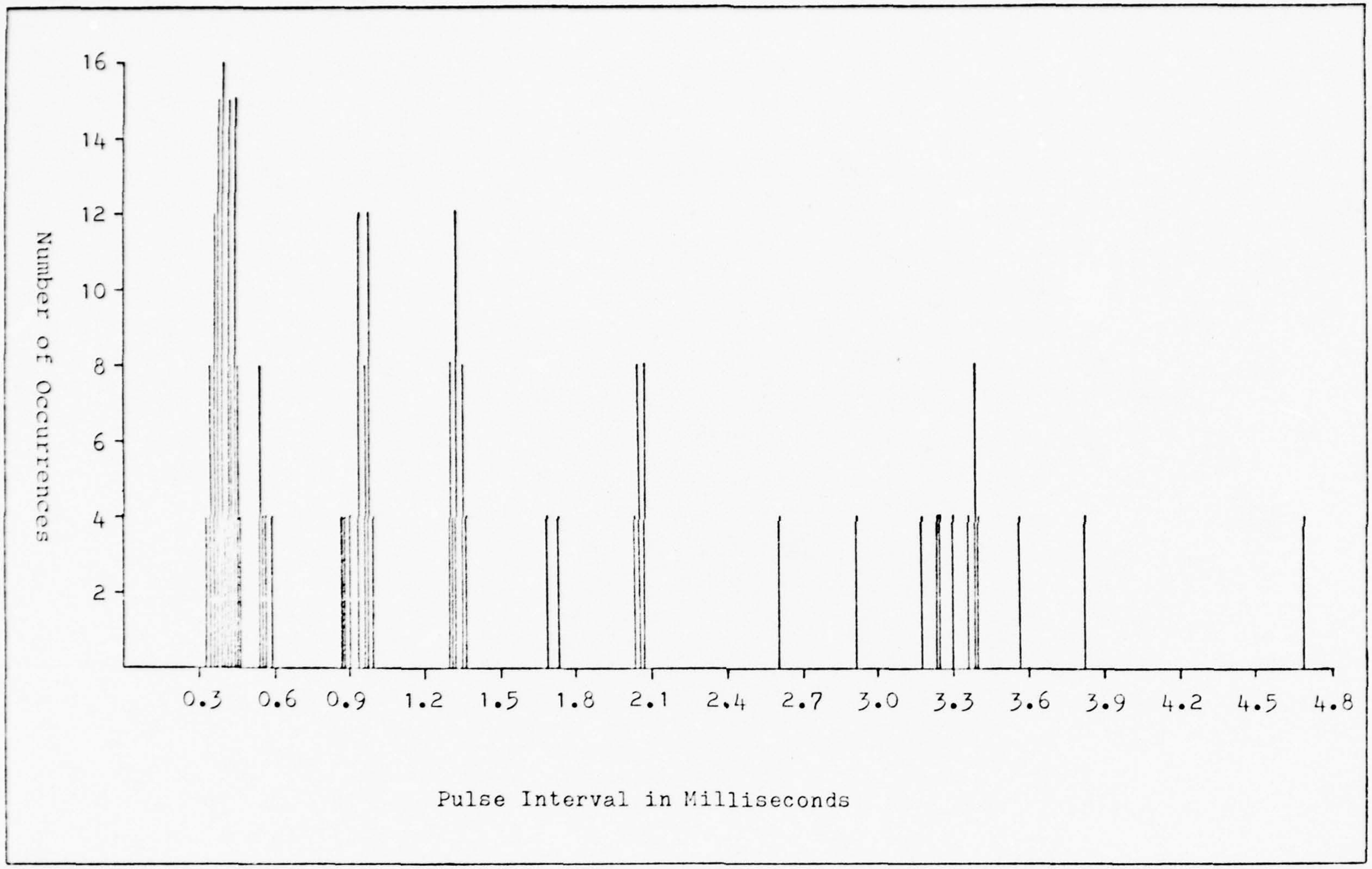


Fig. 24. Pulse Interval Histogram of Synthetic IY

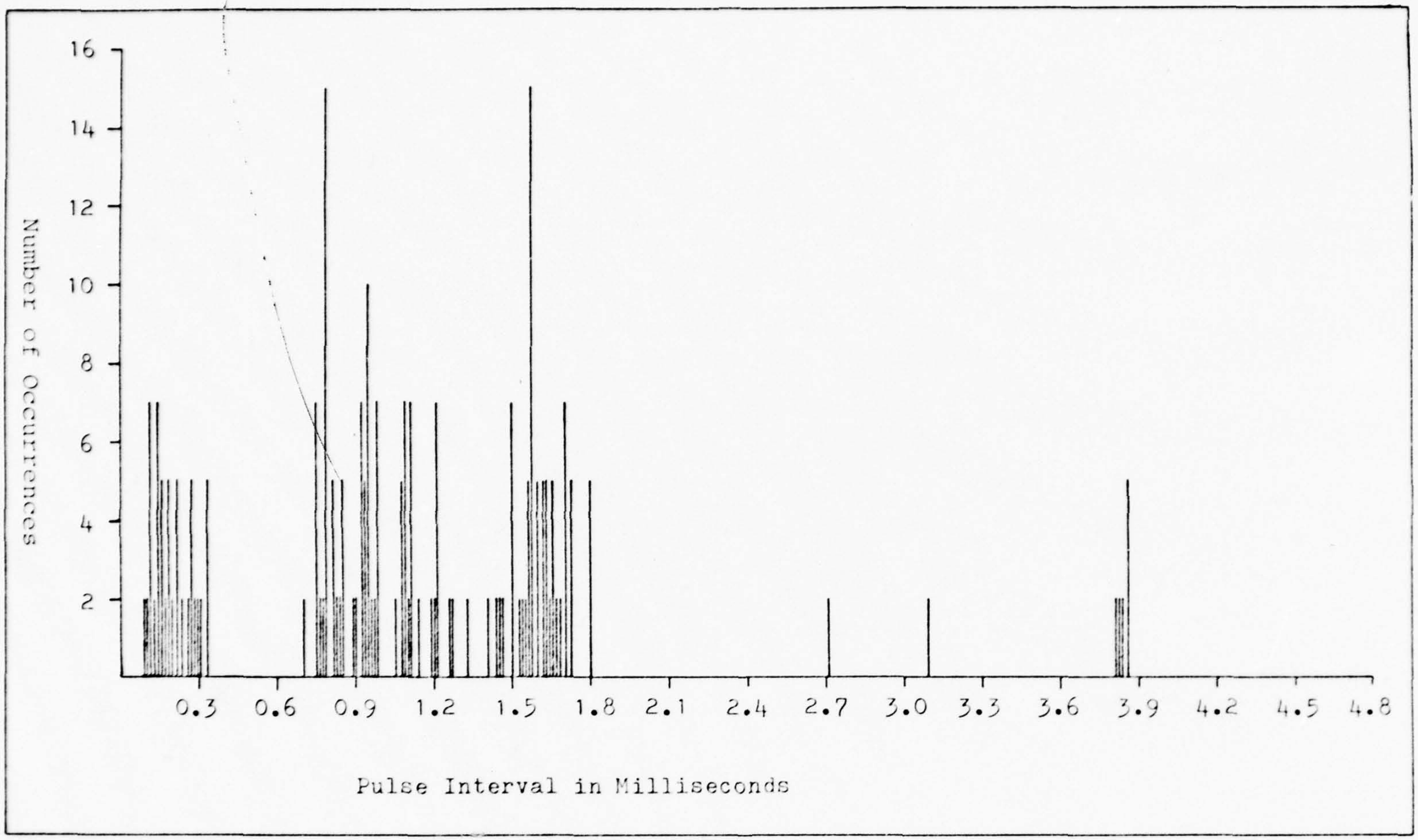


Fig. 25. Pulse Interval Histogram of Synthetic AA

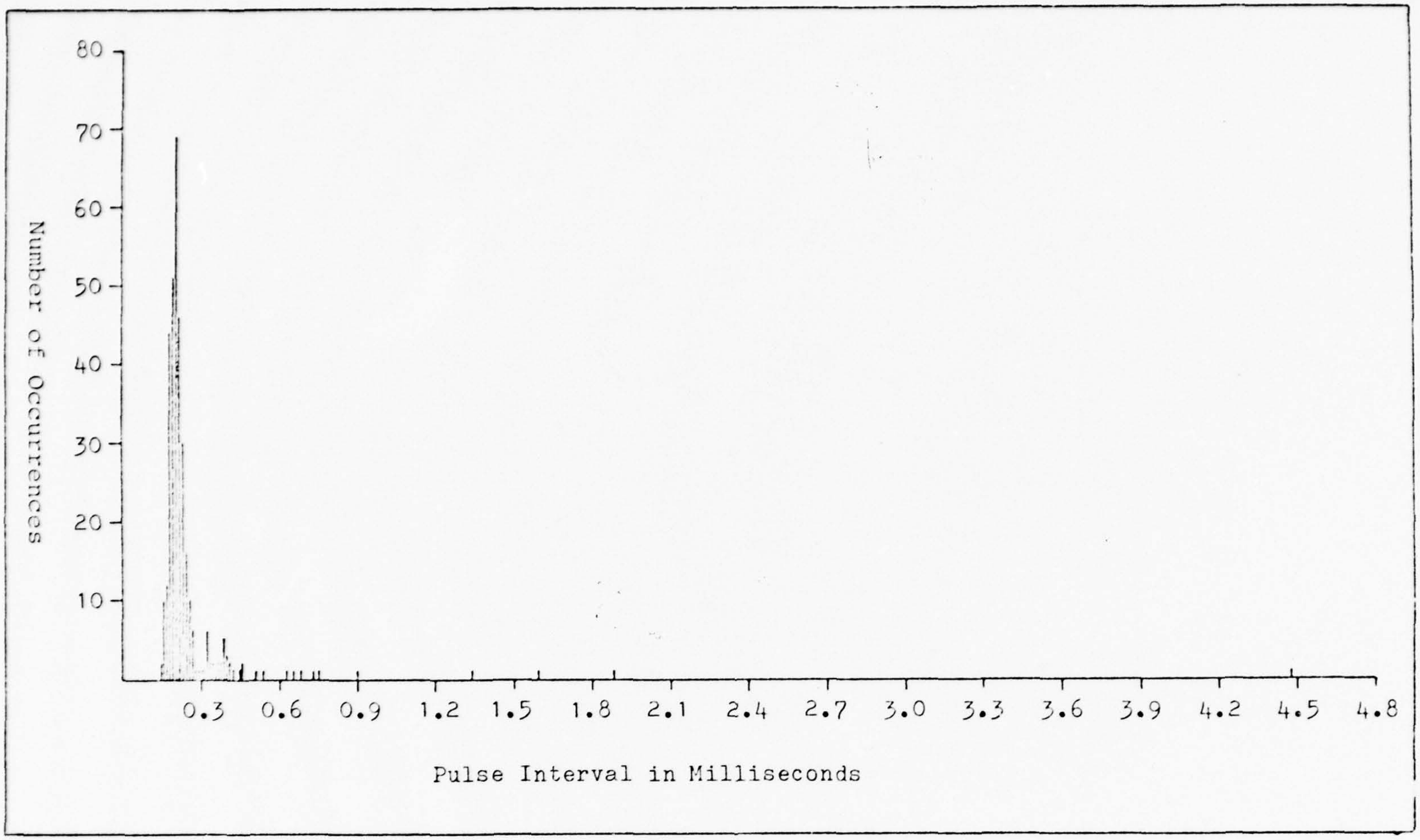


Fig. 26. Pulse Interval Histogram of Synthetic SS

utterance initial position and the final "release" portion of the stop may be missing when it is in the utterance final position.

The parameters and characteristics used for recognition of synthesized speech are also present in natural speech. Although natural speech is more complex and less consistent, there is reason to believe that the methods presented here will make an excellent starting point for the recognition of natural speech.

Steady-State Speech

Steady-state speech, in which sounds approach and remain near some "steady-state" patterns, includes the ten vowels, two of the four semi-vowels (LL and RR), three nasals, four voiced and four voiceless fricatives as shown in Table I on page 3. (The other two semi-vowels, YY and WW, start near a particular vowel, IY for YY and OO for WW, and glide toward the following sound.) In all these cases the sounds are treated in an identical manner; the only differentiation between voiced and voiceless is in the manner of segmentation; pitch period for voiced sounds; 10 ms segments for voiceless.

Each segment is examined and identified independently of the preceding and succeeding segments. Final identification is made for each segment based on the results of three independent classification procedures. The first classification is based on three moments which are calculated from the 480 element pulse interval matrix. (A fourth moment is used in partitioning the speech signal into phonemes as discussed in Chapter IV.) The second classification is a "pattern match" of the pulse interval matrix with similar matrices from the 25 masters

(known steady-state sounds). The third classification procedure is a pattern match of the 30 element channel firing matrix with similar matrices of the masters. The results of these three methods are combined to determine the most likely candidate for each segment.

Moments. Four different moments are calculated from the data in the pulse interval matrix. The first of these (referred to here as the raw moment) is a standard first moment about 1.0 ms (1000 Hz). Any pulse interval occurrences of less than 1.0 ms are considered negative, and any pulse interval occurrences greater than 1.0 ms are considered positive. Thus, a sound with a high incidence of short intervals (high frequency) would have a large negative raw moment and a sound with a high incidence of long intervals would have a large positive raw moment. The raw moment is used in partitioning speech into phonemes.

For the other three moments, the pulse interval matrix is divided into three overlapping sections which correspond, more or less, to the frequency regions of the first three formants. These sections are 0.01 to 0.54 ms, 0.50 to 1.40 ms, and 1.20 to 4.80 ms. A standard first moment is calculated about the short interval (high frequency) end of each of these sections. Each segment is scored against similar measures from the reference sounds by calculating the Euclidean Distance (square root of the sum of the squares of the differences) between the three moments of the incoming signal and the moments of the 25 masters.

Pattern Matches. The pulse interval matrix and the channel firing matrix for each segment of the incoming signal are correlated against the corresponding matrices of the reference sounds. The correlations are performed by using the following equation:

$$K = \frac{\sum_{n=1}^{j} V_{sn} V_{in}}{\sqrt{\sum_{n=1}^{j} V^2_{sn} , \sum_{n=1}^{j} V^2_{sn}}} \qquad (1)$$

where K = correlation factor

V_{sn} = nth element of the standard pulse interval or channel firing matrix

V_{in} = nth element of the incoming pulse interval or channel firing matrix

j = 480 for the pulse interval correlation; 30 for the channel firing correlation.

Actually, K in the above equation is the square root of the correlation factor. However, because the actual numeric answer is not used and because the square root is also a monotone increasing function, it was decided to dispense with squaring the result. In a normal correlation, K can vary from 1.0 (absolute match) to 0.0 (absolute mis-match) to -1.0 (absolute negative match). In this case, the result can only vary from 1.0 (absolute match) to 0.0 (absolute mis-match) because there cannot be a negative number of occurrences of a pulse interval.

Final Identification of Each Segment. Combining the three classification results to make a final identification for each segment is very simple but appears to be effective. For each method of classification the sound candidates are ranked by the scores which are taken as a measure of how closely their characteristics match those of the incoming signal segment. The possible rank for each sound candidate ranges from 1 for the cadidate that is "closest" to the incoming signal to 25 for the candidate that is "farthest" from the incoming signal segment. The

rank for each master is added across the three methods and the sound candidate with the lowest total ranking is selected as the most likely candidate for that segment.

Stops Internal to the Utterance

It was quickly discovered that the classification methods used for steady-state speech did not work for stops. One complication was that sounds adjacent to a stop affect the characteristics of the stop. This problem was solved by using up to six variations of each stop as reference patterns. However, an additional problem was observed which was not as easily solved.

The CxC Computer is somewhat sensitive to amplitude and the signal amplitude drops rapidly and increases rapidly during a stop. Consequently, there are few pulses in the low amplitude segments that are of great interest in stops and, when the number of pulses is extremely low, the pulse interval matrix correlation scores and the moments of the pulse interval matrix are more susceptible to minor variations of the input signal. This susceptibility caused the correlation and moments of the pulse interval matrix to be unsuitable measures for stops. Although the correlation of the channel firing matrix did prove to be viable, another method of classification had to be found that was less susceptible to minor signal variations than the moments and correlation of the pulse interval matrix.

A method of classification that is less affected by signal variations is to divide the pulse interval matrix into several overlapping sections or "windows." The dimensions of the windows were selected by

studying a composite histogram for segments in the shut down and release of several stop variations and selecting the null or low points in the histogram. Thus, the peaks in the histograms of the segments used are contained in one or more windows. The limits of the windows are 0.01 to 0.30 ms, 0.20 to 0.48 ms, 0.43 to 0.74 ms, 0.64 to 0.95 ms, 0.90 to 1.20 ms, 1.10 to 1.55 ms, 1.50 to 1.85 ms, 1.83 to 2.13 ms, 2.11 to 2.40 ms, 2.30 to 2.70 ms, 2.60 to 3.10 ms, 3.00 to 3.80 ms, 3.75 to 4.37 ms, and 4.34 to 4.80 ms. For each segment in question, the number of pulse interval occurrences that fall into each window is determined and the result is correlated against standards for all reference stop variations. Reference patterns for both shut down and release segments of several variations of each stop are stored giving a possibility of up to 72 total reference patterns for the six stops. Currently, only 39 different reference patterns are being used. These patterns are from the last shut down or first release segment of a particular stop in which the number of pulses in the segment exceeded 25. One pattern for each of the shut down and release of each of the six stops was stored. These 12 patterns were tested against various sound combinations and when a problem arose an additional pattern was stored.

Unfortunately, even with the use of multiple masters for each stop the correlations of the channel firings and the window functions, in and of themselves, were not sufficient to identify the stops. The patterns for D and G, for example, are very similar to vowels and they correlate very well with the vowels. For synthetic speech, the "correlation factor" for a vowel against a master for D could be as high as 0.90. A master pattern for B, on the other hand, may correlate against a vowel with a result as low as 0.0. Therefore, if a vowel followed by a B is

input to the system, in order to correctly identify the B the correlations against the B masters would have to rise dramatically while the correlations against the D masters would have to fall dramatically. For an example, say a D master gives an average of correlation of 0.80 during the vowel and then falls to a low of 0.65 during the stop. Further, say the best B master gives an average correlation of 0.10 during the vowel and then rises to a high of 0.60 during the stop. Figure 27 on page 61 presents just such an example of window function correlations against a master for each B, D, and G for a vowel-stop-vowel combination. The figure is a plot of correlation against time for the three masters. Only three stop variations are used for clarity. Even at first glance it is fairly obvious that the stop being analyzed is a B. But, it is necessary for the system to automatically make the same determination and simply taking the highest result for any segment is not sufficient. Therefore, a method had to be incorporated that discriminates the rises or falls of the correlation results.

One such method is to select a point during the preceding sound and use it as a baseline for measuring the rise or fall of the correlations. It was decided to use as a measure the ratio of the increase of the correlations for the current segment versus the maximum possible increase above the baseline. If the baseline for a stop master is 0.60 then the correlations can rise, at most, 0.40. If the correlations rise 0.20, then the result of the discrimination function will be 0.20/0.40, or 0.50. The correlation rose one-half of the distance possible. The baselines should be selected during the stable portion (no transitions going on) of the sounds preceding or succeeding the stop. The segments

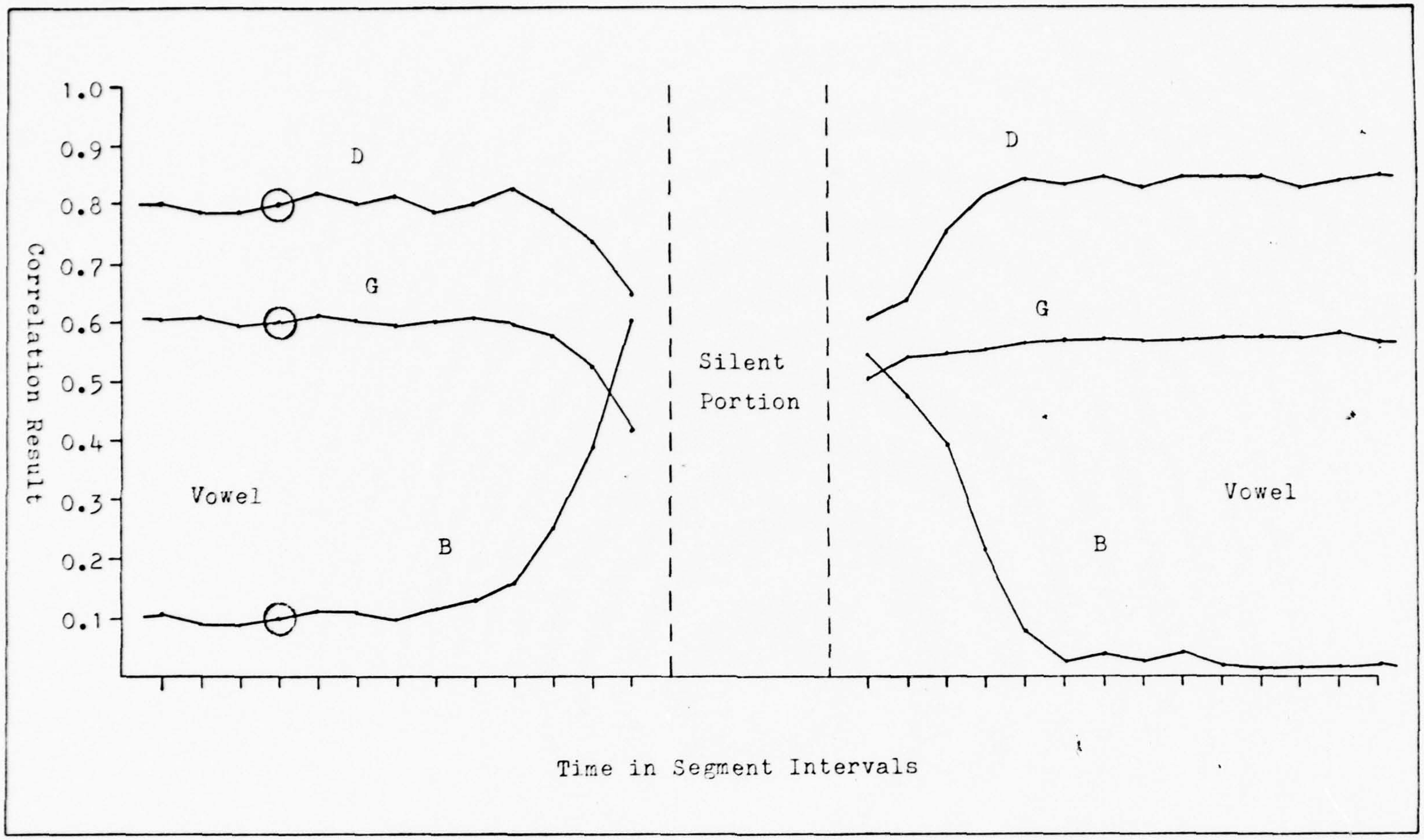


Fig. 27. Window Function Correlations for Vowel-B-Vowel

used for discrimination are slected by the partitioning algorithm as discussed in Chapter IV.

For purposes of an example, say the points circled in Figure 27 on page 61 are selected as the baselines for the three stops. If the following discrimination function is used

$$D_i = \frac{V_{ci}-V_{si}}{1.0-V_{si}} \tag{2}$$

where: D_i = the discriminated result for the window function of the ith stop variation

V_{si} = the baseline for the ith stop variation

V_{ci} = the result of the correlation of the current segment of the incoming signal against the ith stop variation

for each segment of Figure 27 on page 61 the result is as pictured in Figure 28 on page 63. Again it is obvious that a B was being analyzed but this time simply taking the highest correlation result is sufficient. The same procedure is used for the channel firing correlation.

In the above example, the masters used for the shut down of the stop could have also been adequate for the release of the stop but this is not normally the case. Normally the shut down and release of a stop differ greatly and different masters are required for each. Therefore, for a shut down of a stop only the segments just prior to the silent period are examined and for the release of a stop only the segments just after the silent period are examined. For the shut down the baselines from the preceding sound are used and for the release the baselines from the succeeding sound are used.

Each stop may have several reference patterns for the shut down and several reference patterns for the release and there are two

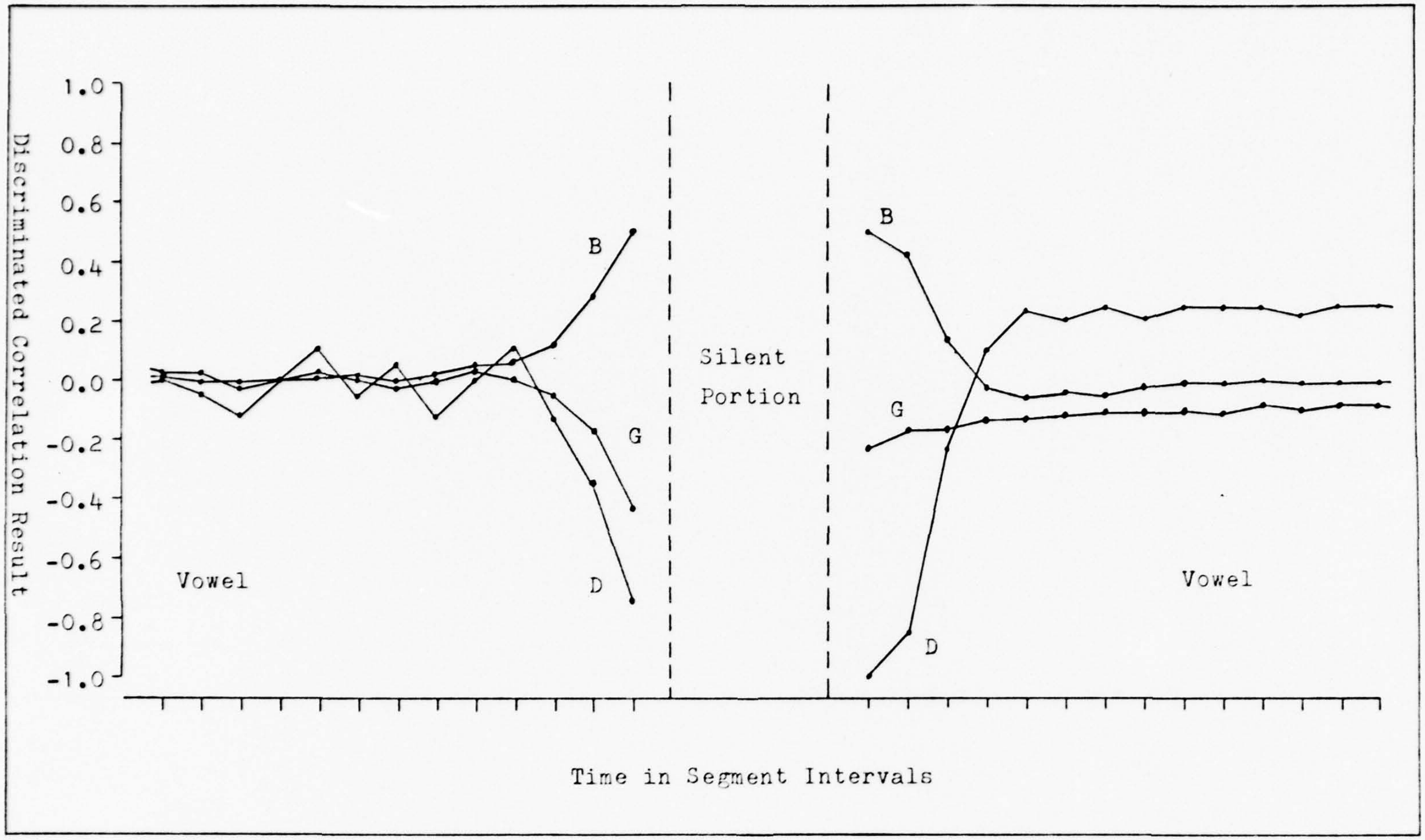


Fig. 28. Window Function Correlations for Vowel-B-Vowel After Discrimination

correlation results for each shut down pattern (window function and channel firing) and two correlation results for each release reference pattern. The highest discriminated result for shut down window function, shut down channel firing, release window function, and release channel firing (total of four) for each stop are added and the stop with the highest sum is selected as the most likely candidate.

Aspirant (H)

The aspirant (H) may only be formed in conjunction with a succeeding voiced sound which is always a vowel or W in American English. The aspirant is formed by placing the vocal tract into position for the succeeding sound and exciting the vocal tract with noise generated by turbulent air flow through half open vocal cords. This whispered portion of the vowel or W lasts for approximately 100 ms before voicing begins without readjustment of the vocal tract. The synthesis system used here recognizes ten basic vowels. Therefore, it can be considered that there are 11 different aspirants. Reference patterns for the 11 sounds are loaded and used as if they were steady-state sounds with one major exception - if one of the aspirants is determined to be the most likely candidate for a particular segment, the most likely non-aspirant is also recorded. The reason for this exception and the way the alternate candidate is used will be discussed in Chapter IV.

Special Cases

The two special cases are a stop in the utterance initial position and a stop in the utterance final position. These are special cases because both the shut down and release portions of the stop may not be present.

The initial portion of an utterance (beginning of sample or after a pause) is treated as if it were the release portion of a stop internal to an utterance. Discriminated correlations against the release reference patterns of the stop variations are performed but the sum of the highest results for at least one of the stops must exceed a threshold. This threshold was more or less arbitrarily set at 1.5 and some experimentation with natural speech should be performed to possibly determine a more suitable value. If the threshold is exceeded, the stop with the highest sum of discriminated correlation results is considered recognized. If the threshold is not exceeded, it is assumed that no stop is present.

A stop in the utterance final position can be formed either with or without a release portion. Frequently a speaker may add a release portion formed with a low level UH to the end of the utterance. However, the release portion is usually very low level and rudimentary. The second method of forming a stop in this position is to terminate the utterance with the closure of the stop. In either case, the release portion of the stop is not available for identification. Therefore, as in stops in the utterance initial position, the final portion of an utterance is treated as part of a stop. Discriminated correlations against the shut down reference patterns of the stop variations are performed but the sum of the highest results for at least one of the stops must exceed a threshold. Again the threshold was set at 1.5. If the threshold is exceeded, the stop with the highest sum of discriminated correlation results is considered recognized. If the threshold is not exceeded it is assumed that no stop is present.

IV. Partitioning and Phoneme Identification

Partitioning an unknown speech sample into useable size pieces is a significant problem for any type of automatic speech recognition. In many automatic speech recognition attempts the analog signal is partitioned into word or phrase size lengths by using silence before and after to demarcate the boundaries. Usually the words or phrases are not recognized as sequences of phonemes, but rather the entire length of signal is treated as a single pattern. This system, on the other hand, individually identifies short analog segments only a few milliseconds in length. These segments are either naturally demarcated by the source or are demarcated by a 10 ms time interval. A voiced speech segment is the result of a single impulsive type excitation of the vocal tract. In either case, the segments are sub-units of phonemes, which are the basic units of speech, and it is necessary to partition and group the sequence of identified segments into phonemic units. The partitioning scheme is based on measures which reflect changes in the speech signal.

Particular points in the speech signal are selected as baselines (starting points) for each of the measures and the changes in the succeeding segments are measured against the baselines. When the distance from the baseline of one of the measures exceeds a threshold, it is considered that a change in the speech signal has been encountered and the current segment becomes the baseline for that measure. When two or more measures indicate a change in the speech signal within three segments of one another, it is considered that a phonemic transition is

taking place, a partition boundary is indicated, and a phoneme is identified.

Individual Partition Measures

Three independent measures are used in this partitioning scheme. They are pulse interval matrix correlations against IY, AA, and OO; the raw moment of the input signal; and the overall input signal amplitude.

The first partition measure is calculated from correlations of the pulse interval matrix of the incoming speech signal against the pulse interval matrices of the masters for IY, AA, and OO. These three vowels were selected because they are generally considered to represent the three corners of the vowel space and most transitions from one phoneme to another will cause a change in the result of the correlation of the incoming signal with at least one of these vowels. The correlation scores with each of these vowels are averaged over three segments in order to "smooth" the parameter and thus filter out most variations within a phoneme. The actual measure is the difference between the current average and a baseline. The results of the correlations for the second segment encountered in an utterance are used as a baseline for the first partition. The Euclidean Distance is calculated from the baseline to the current average for each new segment. When the resulting value exceeds 1.5, it is considered that a change in the speech signal has been encountered and the baseline for this measure is moved to the current average. The process is then repeated.

The second partition measure is based on the raw moment of the incoming signal, as calculated in Chapter III. The raw moment of the current segment is averaged with the raw moments of the preceding two

segments for smoothing. The measure is the difference between the current average and a baseline. Again the baseline is originally the second segment of the utterance. The baseline is subtracted from the current raw moment average. When the absolute value of the result exceeds 3000 it is considered that a change in the speech signal has been encountered and the baseline for this measure is changed to the current average. The process is repeated.

The third partition measure is based on the overall input signal amplitude. Pulses on the amplitude indicator channel (second CxC channel) occur at a rate logarithmically proportional to the overall signal amplitude. The actual calculations are based on the time between the last amplitude marker pulse encountered and the one previous to it. If no amplitude marker pulses are encountered (amplitude marker pulse interval is longer than the segment) within a segment, the amplitude value of the last segment is carried over to the new segment. This measure is also averaged to help filter out local perturbances and again the baseline is originally the second segment of the utterance. It is considered that a change in the speech signal has occurred when the absolute value of the difference between the current amplitude average and the amplitude baseline exceeds 1.5, which corresponds to approximately 4 db. Again, the baseline is moved to the current average and the process repeated.

Partition Boundaries

When two or more partition measures indicate a change in the speech signal within three segments of one another a partition is considered complete and a partition boundary is indicated. However, during

phoneme-to-phoneme transitions speech patterns may change enough within a few segments that boundary conditions may be met several times within a single transition. To prevent multiple boundary markers in such a situation, the partitioning algorithm does not permit two boundary markers to occur within four segments. Further, should the boundary conditions be met within four segments of the last time they were met, not only is a boundary not marked but the boundary marker is inhibited for four more segments. At initial start-up the boundary marker is inhibited for five segments to allow the system to settle down.

Phoneme Identification

Once the partition boundaries are determined, the steady-state phonemes are ranked by the number of times they were recognized at the segment level within that partition. The phoneme which occurred most often is identified as the partition phoneme. If more segments of H were recognized than any other steady-state phoneme, H is identified, but the second most likely phoneme is also recorded. To preclude false identification of a phoneme during a transition, the phoneme identified for a partition must have occurred at least three times. Regardless of whether a steady-state phoneme is identified or not, each time a partition boundary is indicated the tally of phonemes recognized at the segment level is restarted.

When a steady-state phoneme is identified with a particular partition, two checks are made before it is accepted into the final phoneme string output. First, if the phoneme is the same as the last phoneme, a spurious boundary is assumed and the current phoneme is ignored.

Second, if the previous phoneme was an H and the current phoneme is not a vowel or W, the H is replaced by the second mostly likely phoneme for the previous partition.

Combinational Sounds

The phoneme string is also examined to allow the recognition of combinational sounds; that is, sounds that are, or can be thought to be, made up of two phonemes. This category includes the diphthongs and the affricates.

The diphthongs (EI, AI, OI, OU, and AU) are generally thought of as two vowels in tamdem. In both natural and synthetic speech the generator starts at or near the targets for the first vowel and migrates toward the targets of the second targets. Natural speakers do not always reach the second targets. This tendency was also built into the speech synthesis system that was used in this exercise. In EI, AI, and OI the second target is IY but the speaker (real or synthetic) may only reach its closest neighbor, II. Therefore, the first sounds (EE, AA, and OW) in combination with II or IY must be considered complete diphthongs.

The affricates (CH and J) are synthesized by combining T and SH for CH, and D and ZH for J. In this recognition system whenever these combinations are encountered, the appropriate affricate is identified. The affricates are one area in which the recognition of synthetic speech may differ greatly from that of natural speech. Because the affricates have a low frequency of occurrence in American speech - CH appears about 0.44% of the time and J appears about 0.52% of the time (Ref. 2:5) - no major effort was expanded on this aspect of natural speech.

The phoneme string is examined and if the current phoneme is possibly the second phoneme of a diphthong or affricate, the previous phoneme is checked to see if it is the first part. If it is, the two are replaced by the appropriate diphthong or affricate. Also, if the current phoneme is the second part of a diphthong or affricate and the previous phoneme is that diphthong or affricate, the current phoneme is ignored.

Stop Discrimination

As noted in Chapter III, the values of the correlations against the various stop masters must be discriminated. Discrimination is done by normalizing the correlation results during a stop to the correlation results for segments that are assumed to be part of the stabilized portion of the preceding and succeeding sounds. It is assumed when boundary conditions have not been met for four segments (boundary marker is no longer inhibited) that the signal parameters have more or less stabilized and the current segment can be used for normalizing the stop correlations.

When signal parameters are considered stabilized for the first time in an utterance, a test is made to see if the utterance began with a stop. The stop correlations for the release portion of the various stop reference patterns are normalized and the highest correlations (window functions and channel firing) for each stop are added. If the sum for any of these exceeds 1.5, a stop is assumed to be present and is identified as the stop with the highest of these sums. Whether a stop is recognized or not, the shut down portion correlations are

recorded in case the next phoneme is determined to be a stop and they are needed for normalization.

If a silent period in excess of 35 ms is detected during a partition internal to the utterance, it is assumed that a stop is present but the system continues until the signal parameters are assumed to have stabilized in the next partition. If a stop is considered to be present, the shut down portion of the stop correlations are normalized by the recorded values from the last partition and the release portions of the stop correlations are normalized by the results of the correlations against the current segment. The four correlations (shut down and release of both window functions and channel firing) for each stop variation are summed. For a stop internal to the utterance there is no threshold requirement and the stop with the highest sum is identified. If, on the other hand, a stop is not considered to be present, a stop is not identified. Either way, the current correlations against the shut down portions of the various stop masters are recorded in case the next phoneme is determined to be a stop.

When the end of an utterance is encountered, a test is made to see if it ended with a stop. The shut down portion reference correlations are normalized by the values recorded from the last partition and are added for each stop variation. If any of these sums exceeds 1.5, a stop is considered to be present and is identified as the stop with the highest such sum.

V. Evaluation, Results and Recommendations

Evaluation

The purpose of this dissertation was to produce a system that would accept the acoustic output of a particular speech synthesis system and produce an accurate written representation of the input. In all cases, the parameters or characteristics used in the recognition of the synthetic speech are believed to also be present in natural speech. Occasionally some natural speech analysis was performed along with the analysis of the synthetic speech. However, evaluation of system performance during development was done on isolated synthetic phonemes whenever possible. The overall accuracy of the recognition of synthetic steady-state phonemes in isolation (unconnected) was excellent. The system did make occasional errors on individual segments but rarely misidentified or missed a steady-state phoneme. Obviously, development and evaluation of the stops (B, D, G, P, T, and K) and the aspirant (H) had to be done in combination with other phonemes because these phonemes cannot occur in isolation and because the adjacent sounds are known to affect the characteristics of these sounds. The system accuracy on stops and H in "isolation" was very good.

Phonemes rarely occur in isolation in speech; more often they occur in connected sequences to form words and phrases. Testing of overall system performance was performed on isolated words which permitted evaluation of the phoneme based recognition system with connected phoneme strings but stopped short of requiring development of word boundary rules. The word lists used in the tests were developed by the Central Institute for the Deaf (CID) and are phonemically

balanced lists. The frequency of occurrence of the various phonemes in each list approximates the frequency of occurrence in American speech.

Results

Two considerations that were used in analyzing the results of the system tests on the CID word lists should be noted before discussion of the results. First, YY (as in you) is a sound that starts with a short IY (as in he) and then glides toward the next sound. A separate YY is necessary in speech synthesis but is almost impossible to distinguish from a short IY in speech recognition. Therefore, YY was deleted as a possible candidate and a recognized IY for a YY was considered correct. The second consideration was that the difference between an RR (as in run) and an ER (as in her) is so small that they can almost be considered a single phoneme. Therefore, recognition of one for the other or a sequence of one and then the other was considered correct.

The output of the system is a segment-by-segment printout and a printout of the final phoneme string after the data for the utterance is fully processed. Figure 29 on page 75 is a typical system output. The segment-by-segment printing is one line containing the most likely candidate, the partition marker, the partition inhibit value, and the raw moment. A partition boundary is indicated by setting the partition marker to one. The partition marker is inhibited as long as the partition inhibit value is greater than or equal to zero. The raw moment is included merely as a gross indication of the stability of the input speech signal. If an HH is identified for a particular segment, the second most likely candidate is printed and an HH is printed on the next line. The system also outputs a printout of the highest correlations

Most Likely Candidate	Partition Marker	Partition Inhibit Value	Raw Moment

CODE FOR THIS SAMPLE IS: DDAI L2W9

TH	0	4	2340.
[illegible]	0	3	[illegible]
EE	0	2	[illegible]
[illegible]	0	1	[illegible]
AA	0	0	8176.

	WINOUT	CHOUT
BB	0.91719	0.73035
DD	0.99515	0.[illegible]
GG	0.91037	0.72778
PP	0.55032	0.64778
TT	0.88164	0.54624
KK	0.88949	0.77027

AA	0	-1	5541.
AA	0	-1	2882.
[illegible]	[illegible]	[illegible]	[illegible]
II	0	-1	6199.
II	0	-1	4810.
II	0	-1	7518.
II	1	3	12356.
	HH		
II	0	3	12094.
IY	0	3	20000.
	HH		
ZH	1	3	0.

	WININ	CHIN
BB	0.15231	0.20570
DD	0.37659	0.00000
GG	0.30[illegible]	0.11[illegible]
PP	0.56402	0.00000
TT	0.6[illegible]7	0.7[illegible]
KK	0.20[illegible]7	0.0[illegible]

FINAL PHONEME STRING IS:

DD AI

Fig. 29. Typical System Output /die/

of the window functions and channel firings for each stop when a stop is, or may be, present. The correlations are printed when the partition inhibit value reaches zero for the first time (initial stop possible), when it reaches zero after a silent period greater than 35 ms has been found (internal stop), and when the end of data is reached (final stop possible). After the data for an utterance is processed, the system outputs the final phoneme string formed as a result of the analysis of the utterance.

The method of system evaluation was to compare system output with known inputs, namely, the phonemic input of the CID word lists to the synthesizer. Tables II and III on pages 77 through 80 show the words used, the phonemic spelling used, and the system output for the two word lists used. System errors are underlined. There were a total of 281 phonemes input, of which 245 were correctly identified, 23 were mis-identified, 13 were missed entirely, and 11 were added (Table IV on pages 81 and 82). The sum of the mis-identified, missing and added, divided by the total input, gives a simple error rate of 16.7%. However, many of the errors are predictable or understandable and may be overcome at a higher (word or phrase) level. Figures 33 through 61 on pages 92 through 122 present the segment-by-segment printouts of all words which contained errors.

Error Analysis

Some errors in phoneme identification occurred even though the sequence of segment identifications was likely correct. These errors are involved with the trajectories (movement) of the speech through the

TABLE II

CID Phonemically Balanced Word List One

Word		Phonemic Spelling	System Output
1.	ace	EISS	EISS
2.	ache	EIKK	EIKK
3.	an	AENN	AENN
4.	as	AEZZ	AEZZ
5.	battle	BBAETTLL	BBAETTLL
6.	bells	BBEELLZZ	BBEELLZZ
7.	carve	KKAARRVV	__AARRTH
8.	chew	CHOO	CHOO
9.	could	KKUUDD	KKUUDD
10.	dad	DDAEDD	__AEDD
11.	day	DDEI	DDEI
12.	deaf	DDEEFF	DDEEFF
13.	earn	ERNN	ERNN
14.	east	IYSSTT	IYSSTT
15.	felt	FFEELLTT	FFEEUULLII__
16.	give	GGIIVV	GGIIVV
17.	high	HHAI	HHAI
18.	him	HHIIMM	__IIMM
19.	hunt	HHUHNNTT	HHUHNNTT
20.	isle	AILL	AILL
21.	it	IITT	IITT
22.	jam	JJAEMM	DDEEMM
23.	knees	NNIYZZ	NNIYZZ
24.	law	LLOW	LLOW
25.	low	LLOU	LLOU

TABLE II (Con't)

CID Phonemically Balanced Word List One (Con't)

Word		Phonemic Spelling	System Output
26.	me	MMIY	MMIY
27.	mew	MMYYOO	MMIY<u>RR</u>OO
28.	none	NNUHNN	NN<u>RR</u>NN
29.	not	NNAATT	NNAATT
30.	or	OURR	<u>HHOO</u>RR
31.	owl	AUWWLL	AA<u>RR</u>OO__LL
32.	poor	PPOURR	PPOURR
33.	ran	RRAENN	<u>DDUUEE</u>NN
34.	see	SSIY	SSIY
35.	she	SHIY	SHIY
36.	skin	SSKKIINN	SSKKIINN
37.	stove	SSTTOUVV	SSTTOUVV
38.	them	TEEEMM	TEEEMM
39.	there	TEEERR	TEEERR
40.	thing	THIINNGG	THIINNGG
41.	toe	TTOU	TTOU
42.	true	TTRROO	TTRROO
43.	twins	TTWWIINNSS	TTWWIINNSS
44.	up	UHPP	UHPP
45.	us	UHSS	UHSS
46.	wet	WWEETT	WW<u>II</u>__
47.	what	HHWWUHTT	HH__<u>UU</u>TT
48.	wire	WWAIRR	<u>HH</u>AIRR<u>DD</u>
49.	yard	YYUHRRDD	IY__RRER__
50.	you	YYOO	IYOO

TABLE III

CID Phonemically Balanced Word List Two

	Word	Phonemic Spelling	System Output
1.	ail	EILL	EILL
2.	air	EERR	EERR
3.	and	EENNDD	EENNDD
4.	been	BBIINN	BBIINN
5.	by	BBAI	BBAI
6.	cap	KKAEPP	KKAEPP
7.	cars	KKAARRSS	KKAARRSS
8.	chest	CHEESSTT	SHIIEESSTT
9.	die	DDAI	DDAI
10.	does	DDUHZZ	DDUHZZ
11.	dumb	DDUHMM	DDUHMM
12.	ease	IYZZ	IYZZ
13.	eat	IYTT	IYTT
14.	else	EELLSS	EELLSS
15.	flat	FFLLAETT	FFLLAETT
16.	gave	GGEIVV	__IIEIVV
17.	ham	HHAEMM	HHAEMM
18.	hit	HHIITT	__IITT
19.	hurt	HHERTT	KKERRRTT
20.	ice	AISS	AISS
21.	ill	IILL	IILL
22.	jaw	JJOW	DDHHUUOW
23.	key	KKIY	KKIY
24.	knee	NNIY	NNIY
25.	live	LLIIVV	LLIIVV

TABLE III (Con't)

CID Phonemically Balanced Word List Two (Con't)

Word		Phonemic Spelling	System Output
26.	move	MMOOVV	MMOOVV
27.	new	NNOO	NNOO
28.	now	NNAU	NNAU
29.	oak	OUKK	OU__
30.	odd	AADD	AADD
31.	off	OWFF	OWFF
32.	one	WWUHNN	WWUHNN
33.	own	OUNN	OUNN
34.	pew	PPYYOO	TTVVOO
35.	rooms	RROOMMSS	DDMMOONNSS
36.	send	SSEENNDD	SSEENNDD
37.	show	SHOU	SSUUOU
38.	smart	SSMMAARRTT	SSMMAARRIITT
39.	star	SSTTAARR	SSTTAARR
40.	tear	TTIIRR	TTIIRR
41.	that	TEAETT	THEETT
42.	then	TEEENN	__EENN
43.	thin	THIINN	THIINN
44.	too	TTOO	TTOO
45.	tree	TTRRIY	TTRRIY
46.	way	WWEI	MMIIIY
47.	well	WWEELL	WWEELL
48.	with	WWIITH	WWIIHH
49.	young	YYUHNNGG	IYTHRRNNGG
50.	your	YYUURR	IYUUER

TABLE IV

Recognition Statistics

Phoneme	Total Number	Total Correct	Number Added	Totally Missed	Mis-Identified As
IY	10	10	-	-	---
II	13	13	4	-	---
EE	13	12	-	-	II
AE	10	7	-	-	EE, EE, EE
AA	6	6	-	-	---
UH	10	6	-	1	RR, RR, UU
UU	2	2	3	-	---
OO	9	9	-	-	---
OW	3	3	-	-	---
ER	2	2	-	-	---
EI	6	5	-	-	IIIY
AI	6	6	-	-	---
OI	0	-	-	-	---
OU	8	7	-	-	HHOO
AU	2	2	-	-	---
WW	9	5	-	2	MM, HH
LL	13	13	-	-	---
RR	16	14	2	-	UU, MM
YY	6	5	-	-	VV
MM	10	9	-	-	NN
NN	22	22	-	-	---
NG	0	-	-	-	---

TABLE IV (Con't)

Recognition Statistics

Phoneme	Total Number	Total Correct	Number Added	Totally Missed	Mis-Identified As
VV	6	5	-	-	TH
TE	4	2	-	1	TH
ZZ	5	5	-	-	---
ZH	0	-	-	-	---
FF	4	4	-	-	---
TH	3	2	-	-	HH
SS	15	15	-	-	---
SH	2	2	-	-	---
CH	2	1	-	-	SH
JJ	2	-	-	-	DD, DDHH
HH	7	4	-	2	KK
BB	4	4	-	-	---
DD	12	10	2	2	---
GG	4	3	-	1	---
PP	4	3	-	-	TT
TT	23	21	-	2	---
KK	8	6	-	2	---
Totals	281	245	11	13	23

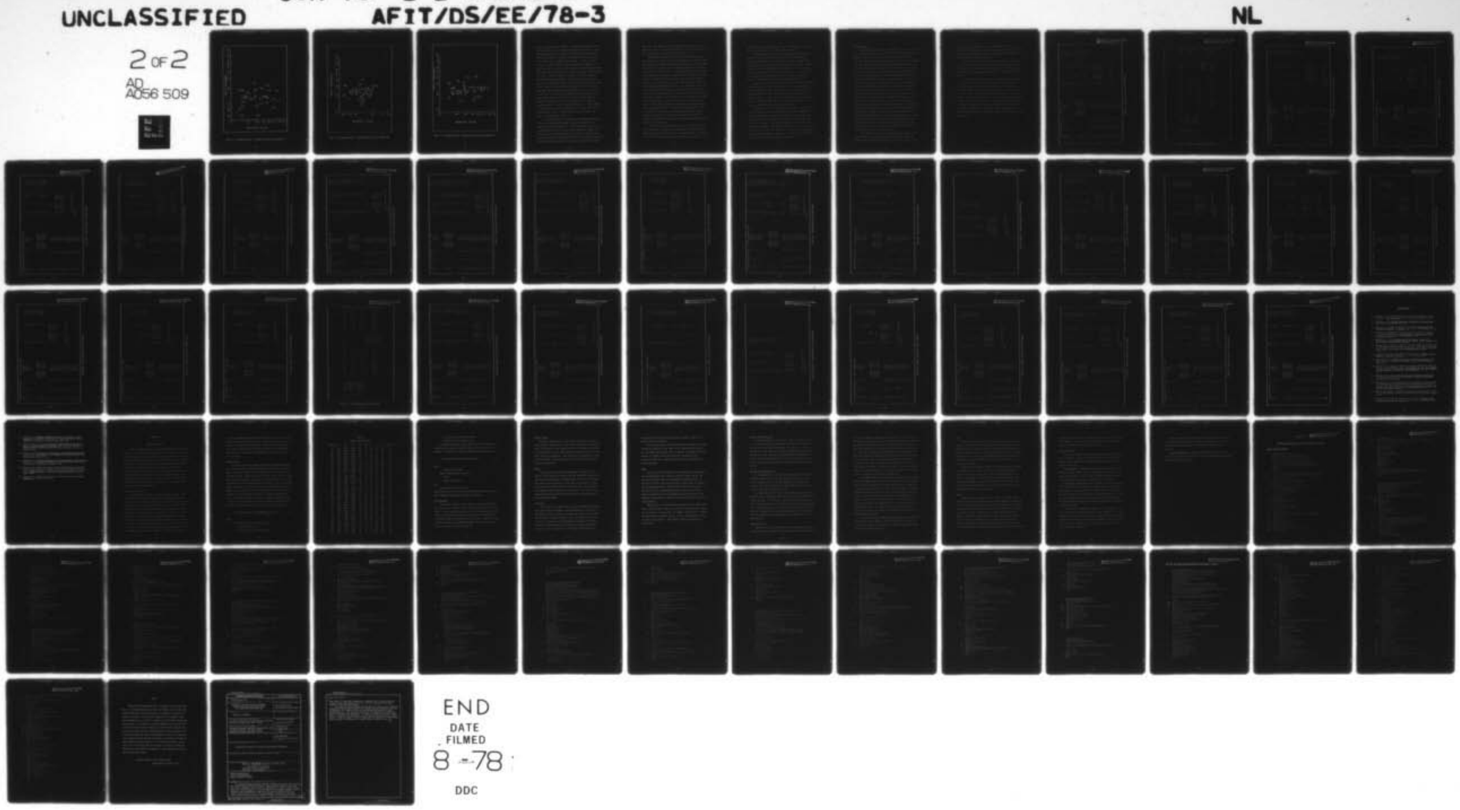
speech pattern space. In order to more easily visualize the problem of trajectories in the speech space, the concept of formant targets must be presented. Every speech sound can be thought to have formant targets associated with it. In the case of voiced sounds (vowels, semi-vowels, nasals, and voiced fricatives), the formants actually migrate from the previous sound to the appropriate targets. In the case of a voiced sound followed by a fricative or stop, the formants move toward the appropriate targets but voicing stops (for fricatives) or the amplitude drops (for stops) prior to the arrival at the targets. In the case of a fricative or stop followed by a voiced sound, the formants move away from the targets toward the voiced sound but voicing starts or amplitude rises after the formants have left the original targets. The targets for stops and fricatives are referred to as virtual targets. In our synthesis system the formants move from one target to another in a manner that can be modeled as an exponential function. That is, they move rapidly away from the locus of the previous sound but slow down as they approach the targets of the succeeding sound (see Appendix A). Figure 30 on page 84 is a formant one versus formant two plot of the formant targets of the various sounds. Figures 31 and 32 on pages 85 and 86 are similar plots for formant three versus formant two and formant three versus formant one, respectively.

Several of the system errors noted in Tables II and III on pages 77 through 80 are thought to be a result of a combination of the current algorithms and the trajectories of the sounds in the speech space. (NOTE: in the following cases all words in the CID word lists will be referred to by list number and word number in a shorthand notation. For example, list two word one would be referred to as L2W1.) In

AD-A056 509 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 6/4
AUTOMATIC RECOGNITION OF SYNTHETIC SPEECH USING AN ELECTRONIC M--ETC(U)
JUN 78 D B WARMUTH

UNCLASSIFIED AFIT/DS/EE/78-3 NL

2 OF 2
AD
A056 509



END
DATE
FILMED
8 -78
DDC

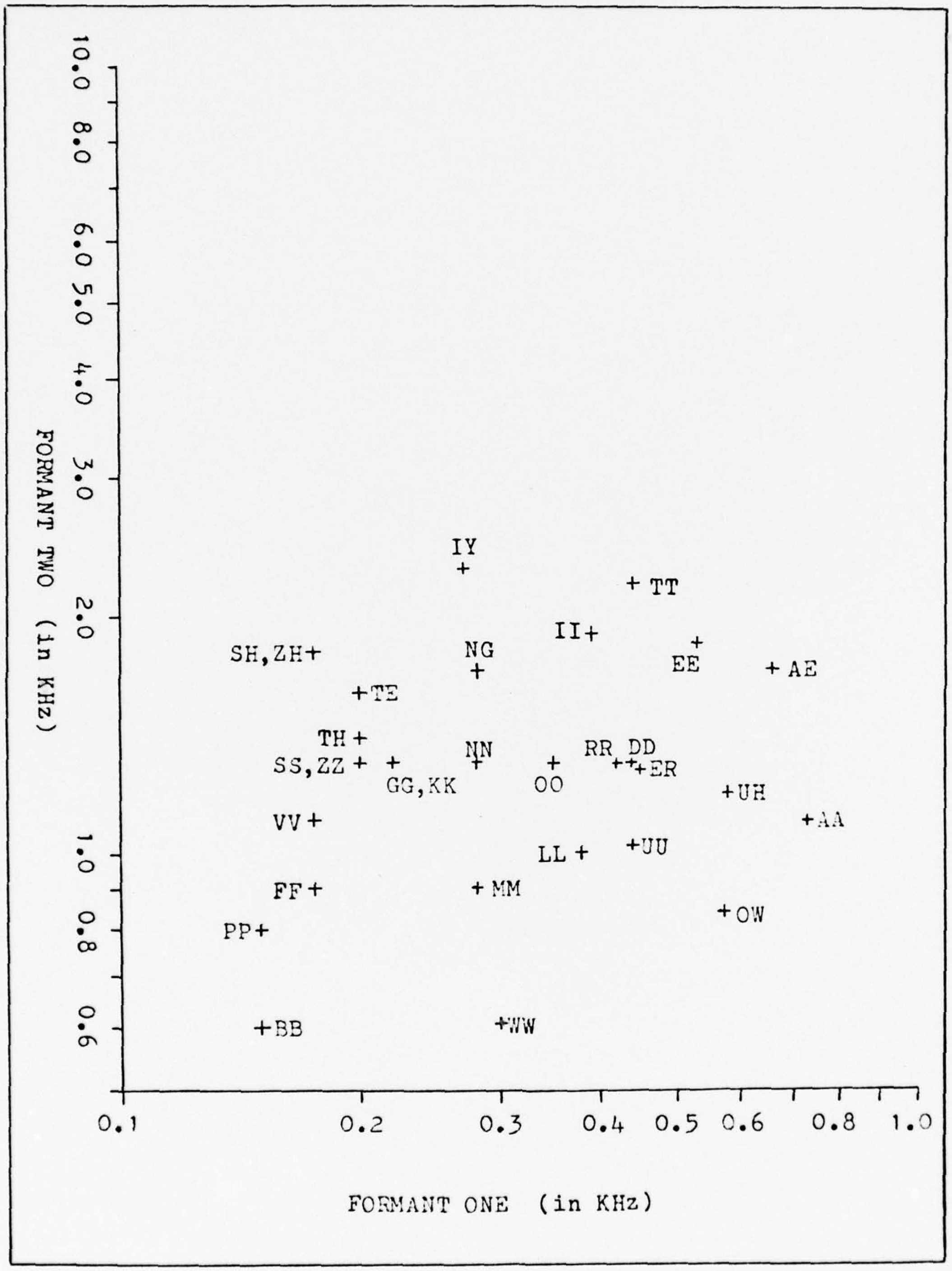


Fig. 30. Formant Targets - Formant Two Versus Formant One

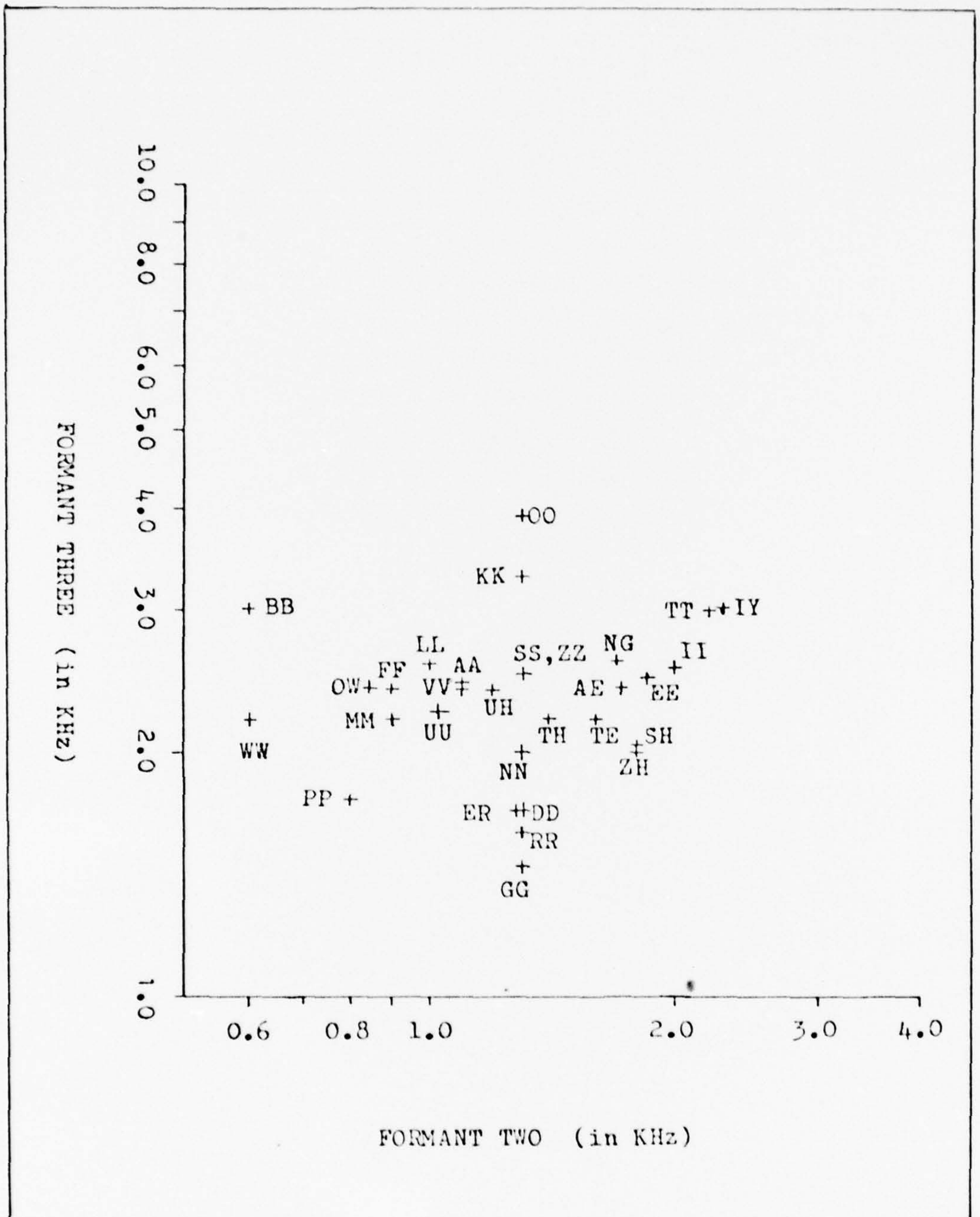


Fig. 31. Formant Targets - Formant Three Versus Formant Two

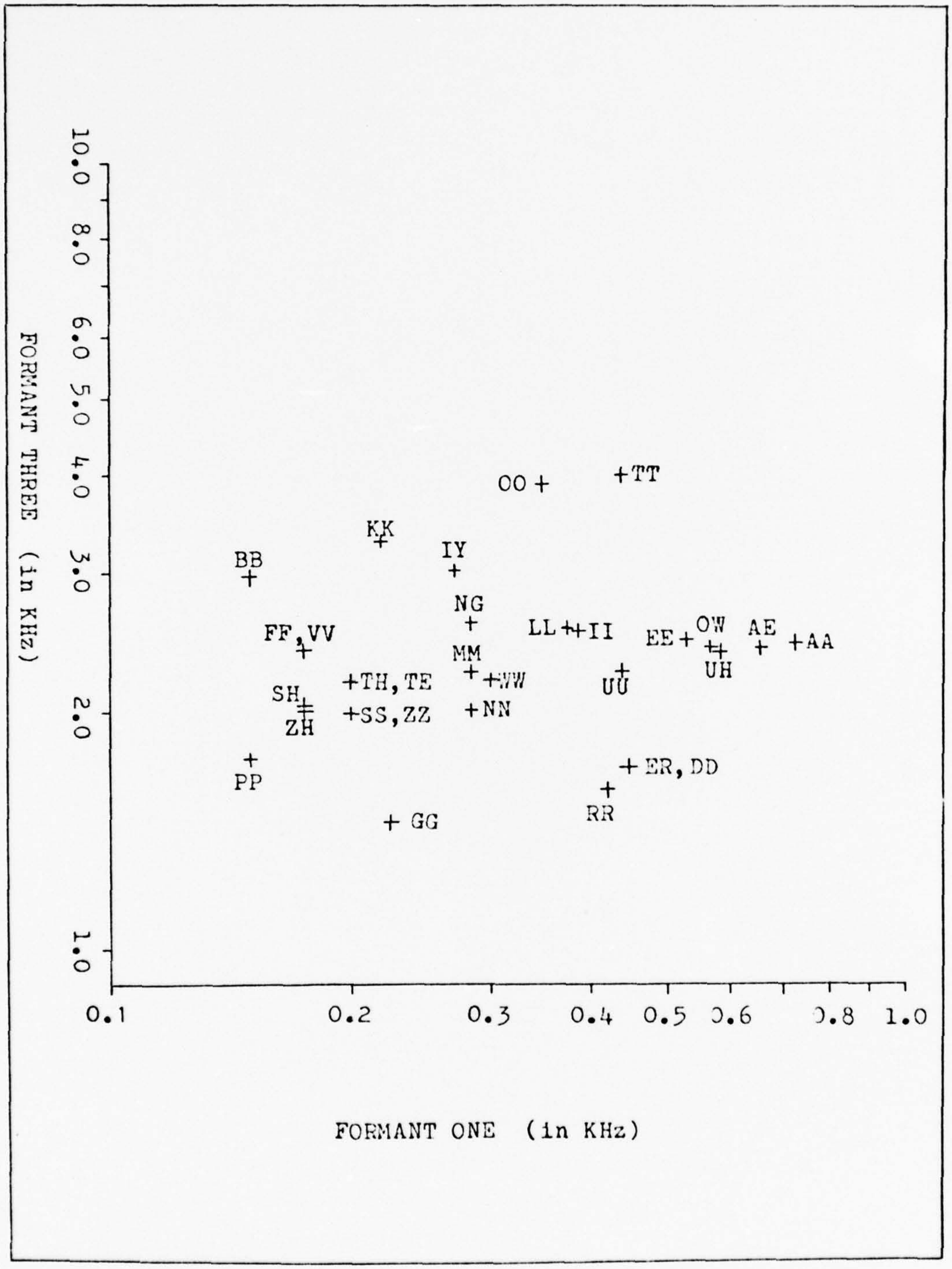


Fig. 32. Formant Targets - Formant Three Versus Formant One

almost all test words the sequences of segment identifications clearly show the migration of the patterns through the speech space. In most cases this migration did not present a problem; the partition marks divided the sequences of segments into groups such that a few simple rules yielded correct phoneme identifications. In other cases we believe it caused mis-identification and additions of phonemes. In L1W22 /jam/ (note that JJ is actually synthesized as DDZH) the movement from ZH through AE to MM produced a sequence of segment identifications of six II's, six EE's, eight AE's and three EE's in one partition. Referring to Figures 30 through 32 which depict the speech pattern space, it is reasonable to expect that the sound moved through the II's and EE's on the way to AE and back through the EE's on the way out toward MM. This progression is understandable; but, EE was selected in preference to AE because the sound with the largest number of segment identifications in a partition is selected as the partition phoneme. Obviously this rule is too simplistic. The segment-by-segment printout for /jam/ is displayed in Figure 37 . Other examples of this problem are seen with the same target in L1W33 /ran/ (Fig. 42) and L2W41 /that/ (Fig. 57) and with other targets in L1W49 /yard/ (Fig. 46) and L1W46 /wet/ (Fig. 43).

An extra phoneme may be recognized if a spurious boundary marker occurs as the sound passes through a transitional phoneme. For example, in L1W15 /felt/ (Fig. 35), as the speech moved from EE to LL it passed near UU as the trajectory slowed, and three segments of UU were identified. During the transition, criteria for partition boundaries were met twice rather than once and the UU was added to the final phoneme string because the three UU segments were found in the extra

partition. Other examples of this problem are L1W27 /mew/ (Fig. 38), L2W8 /chest/ (Fig. 47), L2W16 /gave/ (Fig. 48), L2W22 /jaw/ (Fig. 51), L2W37 /show/ (Fig. 55), and L2W49 /young/ (Fig. 61).

Another type of error observed is the nearest neighbor error; each segment of a string is incorrectly identified as a nearby neighbor and the phoneme is consequently identified incorrectly as that neighbor. This error is frequently observed in human listening panels evaluating natural speech. In that case it is not known whether the error is made by the speaker or the listener. However, in our synthesized speech we are quite certain that the proper targets were used in the synthesis strategy. Yet in L1W28 /none/ (Fig. 39) and L2W49 /young/ (Fig. 61) the sound UH has been identified as RR. It is interesting to note that in both of these cases the vowel is associated with a nasal; in the first case it is surrounded by nasals and in the second it is preceded by a sound somewhat near the nasal NN in the pattern space and followed by NN. Other examples of this type of error are seen in L1W33 /ran/ (Fig. 42) where RR is identified as UU, again in association with a nasal, and L2W46 /way/ (Fig. 59) where EE is identified as II. The fact that the system makes errors in these situations where human observers are also quite likely to make similar errors, lends some credence to the claim that the speech recognition system simulates real auditory system functions.

In four cases an equal number of correct and incorrect segments were found in the same partition and the incorrect phoneme was chosen simply on the basis of an arbitrarily assigned precedence. In L1W30 /or/ (Fig. 40), HH was chosen over OW; in L1W48 /wire/ (Fig. 45), HH

was chosen over WW; in L2W19 /hurt/ (Fig. 50), ER was chosen over HH; and in L2W46 /way/ (Fig. 59), MM was chosen over WW.

Some sounds are characteristically low in amplitude and consequently produce few pulses in the CxC feature extraction process. Segments with fewer pulses are more prone to incorrect identification. Low amplitude of the speech could possibly be the cause of the identification of the final TH as HH in L2W48 /with/ (Fig. 60), the identification of the initial TE as Th in L2W41 /that/ (Fig. 57), in the failure to find the TE in L2W42 /then/ (Fig. 58), and HH in L1W18 /him/ (Fig. 36), L2W18 /hit/ (Fig. 49) and L2W19 /hurt/ (Fig. 50).

Incorrect rules in the synthesizer strategy most probably produced the extra vowel II in L2W38 /smart/ (Fig. 56). The long sequence of II and IY segments in this word is clearly not to be expected in normal speech and probably results from an incorrect transition time in the synthesis strategy. Synthesizer errors are also suspect in L1W33 /ran/ (Fig. 42) and L1W48 /wire/ (Fig. 45) where the RR probably starts and stops so abruptly that the stop DD is identified.

In L1W15 /felt/ (Fig. 35) and L1W46 /wet/ (Fig. 43) the final TT is missed by the stop correlation procedure. However, study of the segment identification sequence reveals that both of these sequences end with four or five segments of II followed by two FF segments and one or two SS segments. This sequence appears to be quite consistent for a final TT phoneme as can be seen in L2W19 /hurt/ (Fig. 50), L2W38 /smart/ (Fig. 56) and L2W41 /that/ (Fig. 57), and suggest that additional experiments should be designed to see if other stops might produce similarly distinctive sequences of segments.

Recommendations

All algorithms and all pattern characteristics used in developing this system are very general and do not make use of attributes that are unique to speech. It is believed that the cause for this fact is two-fold. First, it may be because the author is an Electrical Engineer and not a specialist in speech production or hearing. Second, it may be because characteristics unique to the speech synthesizer absolutely were not to be used and the author may have gone overboard in this area. There are at least three areas where characteristics unique to speech could probably be used with considerable benefit. First, there could be a voiced/voiceless determination that could at least reduce the number of candidates for a particular sound. This information is readily available in the computer since pitch period marker pulses normally occur during voicing and are absent during voiceless sounds. However, this information, which is considered by phoneticians to be the most basic and simple feature of speech characterization, is not utilized in the present phonemic identification process. Second, rate, regularity, and amplitude of the first (or last) few pitch periods at the onset (or cessation) of voicing is available information in the computer and would probably be valuable in the identification of stops and HH. Finally, there are probably several other methods of recognizing the presence of a stop that are far superior to the methods developed in this paper and these methods would probably increase the likelihood of correctly identifying the stop.

There could also be some way to monitor the trajectories of the speech signal in some speech space that would resolve many of the problems discussed above. This monitoring could be as simple as the

proper use of the sequences of segment identifications or it could be as complex as calculations based on the actual correlation values of the various phoneme candidates. Or indeed there may be some totally different methods of performing this task.

The system was informally tested on natural speech. The results were somewhat disappointing. Segment identification appeared to be acceptable but it is our opinion that algorithms to identify phonemes from the segment identifications will have to be improved for natural speech.

We have shown that the system is reasonably accurate on synthesized speech as it now stands. The system should be subjected to rigorous testing on natural speech to determine if decision criteria should be modified to improve performance. It must be remembered that contextual information was not used in the development of the current system. We firmly believe that with further research and the addition of some simple phonetic and linguistic rules this system can be developed into a working speech recognizer that requires only a small computer (or a small part of a large one), requires relatively small amounts of processing time, and has the potential of an almost unlimited vocabulary.

CODE FOR THIS SAMPLE IS: KKAARPVV L1W7

ER	0	4	13785.
EE	0	3	8545.
AA	0	2	89.
AA	0	1	5552.
AE	0	0	2814.

HH

	WINOUT	CHOUT
BB	0.47132	0.64246
DD	0.46431	0.60363
GG	0.36755	0.55144
PP	0.54965	0.44993
TT	0.39383	0.47803
KK	0.67375	0.66398

AA	0	-1	4487.
AA	0	-2	2293.
AA	0	-3	1952.
AA	0	-4	1524.
AA	0	-5	2285.
AA	0	-6	1573.
AA	0	-7	1550.
AA	0	-8	2162.
AA	0	-9	1633.
AA	0	-10	2006.
AA	0	-11	1286.
AA	0	-12	1826.
AA	0	-13	1567.
AA	0	-14	1304.
AA	0	-15	6175.
AA	0	-16	3900.
LL	0	3	9827.
LL	1	2	14118.
LL	0	1	10722.
PP	0	0	12324.
PP	1	3	9850.
PP	0	2	10895.
PP	0	1	10028.

PP	0	0	9535.
PP	0	-1	8098.
PP	0	-2	11146.
PP	0	-3	11595.
PP	0	-4	11935.
PP	0	-5	7831.
EP	0	-6	19372.
DD	1	3	20360.
TH	0	3	7192.
TE	0	2	12854.

HH

TH	0	3	7715.
IY	0	2	11285.
TH	0	1	16829.
IY	0	0	2371.
II	0	-1	2310.
YY	0	-2	20000.

HH

ZH	1	3	0.

	WININ	CHIN
BB	0.55059	0.00000
DD	0.07965	0.15109
GG	0.04720	0.16333
PP	0.50437	0.38932
TT	-0.01507	-0.38553
KK	0.00000	0.00000

FINAL PHONEME STRING IS:

AA PR TH

Fig. 33. System Output for L1W7 /carve/

CODE FOR THIS SAMPLE IS: SYNTH DDAEDD L1W10

II	0	4	-3158.
EE	0	3	4500.
[illegible]	0	2	-1025.
[illegible]	0	1	175.
[illegible]	0	0	-2719.

	WINOUT	CHOUT
BB	0.08755	0.[illegible]
DD	0.91197	0.[illegible]
GG	0.02259	0.[illegible]
[illegible]	[illegible]	[illegible]
TT	0.[illegible]	[illegible]
KK	0.00000	0.4[illegible]

EE	0	-1	-1592.
[illegible]	0	-1	-1591.
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	1	3	0.

	WININ	CHIN
BB	0.[illegible]	0.[illegible]
DD	0.[illegible]	0.[illegible]
GG	0.[illegible]	0.17[illegible]
[illegible]	0.[illegible]	0.[illegible]
TT	0.37457	0.60906
KK	0.75[illegible]	0.[illegible]

FINAL PHONEME STRING IS:

AE DD

Fig. 34. System Output for L1W10 /dad/

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
CODE FOR THIS SAMPLE IS:    SYNTH  FFEELLTT  L1W15

FF   [illegible]   [illegible]   -18336.
                                 -19755.
                                 -11498.
                                 [illegible]
                                 -9123.

               WINOUT       CHOUT
EE   0.13227 [illegible]   0.17392 [illegible]
     [illegible]           [illegible]
KK   0.25727 [illegible]   0.44265

[illegible]

               WININ        CHIN
EE   [illegible]           0.40192 [illegible]
     [illegible]           [illegible]
KK   [illegible]           [illegible]

FINAL PHONEME STRING IS:

FF EE LU LL II
```

Fig. 35. System Output for L1W15 /felt/

```
CODE FOR THIS SAMPLE IS:    SYNTH  HHIIMM  L1W18

 II        0        4        2208.          NN        0        2      -3190.
 NS        0        3       22200.                 HH
        HH                                  NN        0  [illegible]   5574.
 FF        0        2       -4834.          [illegible]
 II  [illegible]    3  [illegible]
 II  [illegible]    2  [illegible]
 II  [illegible]       [illegible]
 II  [illegible]    0        -422.
                                            MM        1        3      20000.
                                                    HH
                  WINOUT      CHOUT          ZH        1        3          0.
 BB              0.68380  [illegible]
 DD          [illegible]  [illegible]
 GG          [illegible]  [illegible]
 EE          [illegible]  [illegible]
 TT          [illegible]  [illegible]
 KK          [illegible]     0.64574

 II        0       -1        8180.                WININ       CHIN
 [illegible]                                BB   0.23221  [illegible]
                                            DD  [illegible] [illegible]
                                            GG  [illegible] [illegible]
                                            EE  [illegible] [illegible]
                                            TT  [illegible] [illegible]
                                            KK  [illegible] [illegible]

                                               FINAL PHONEME STRING IS:
        EH                                  II MM
```

Fig. 36. System Output for L1W18 /him/

```
CODE FOR THIS SAMPLE IS:  SYNTH  JJAEMM  L1W22

                      -1772.

          WINOUT      CHOUT
        0.50457     0.67219
        [illegible] [illegible]
        0.57227     0.60752

                      -3766.

 ZH

          WININ       CHIN
 BB     0.25275     [illegible]
 KK     0.34454     [illegible]

 FINAL PHONEME STRING IS:
 DD EE MM
```

Fig. 37. System Output for L1W22 /jam/

```
CODE FOR THIS SAMPLE IS:    SYNTH  MMYYOO  L1W27

                                   20000.

                    WINOUT     CHOUT
                    0.[illegible]     0.[illegible]
                    0.[illegible]     0.[illegible]
                    0.[illegible]     0.[illegible]
                    0.[illegible]     0.[illegible]
                    0.[illegible]     0.[illegible]
                    0.[illegible]     0.[illegible]

                                   20000.

                                                   15376.
                                                   20000.
                                                   20000.
PP        1          2                             20000.
ZH        1          2                                 1.

       WININ      CHIN
EB     0.67577    0.[illegible]
XX     0.[illegible]    0.[illegible]

   FINAL PHONEME STRING IS:
MM IY PP OO
```

Fig. 38. System Output for L1W27 /mew/

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
CODE FOR THIS SAMPLE IS:    SYNTH  NNUHNN  L1W28

                                  338.
                                 -176.
                                 -1588.
                                 -328.

NN         0          0          -1347.

                         WINOUT    CHOUT
                         0.23760   0.88558
                         0.32344   0.87888

                                 -2023.

                    FINAL PHONEME STRING IS:
                  NN EE NN
```

Fig. 39. System Output for L1W28 /none/

```
CODE FOR THIS SAMPLE IS:    SYNTH  DUFR  L1W30
                   4         9554.
                   3         8544.
                   2         9572.
                             11417.
                             9345.

             WINOUT     CHOUT

                  -1         9701.
                  -2         10811.
                  -3         9627.
                             11307.

                             11832.

      WININ      CHIN

FINAL PHONEME STRING IS:
HH OO RR
```

Fig. 40. System Output for L1W30 /or/

```
CODE FOR THIS SAMPLE IS:    AUWWLL   L1W31

AA        0      4            1707.
AA        0      3            2313.
MM        0      2            2653.
MM        0      1            2216.
AA        0      0            4439.


                      WINOUT      CHOUT
BB                    0.22341     0.34736
DD                    0.10268     0.26213
GG                    0.05581     0.12888
PP                    0.10259     0.13586
TT                    0.11164     0.17788
KK                    0.10157     0.26404


AA        0     -1            2385.
[illegible] 0   -2            2631.
[illegible] 0   -3            2605.
[illegible] 0   -4            4432.
[illegible] 0   -5            1102.
[illegible] 0   -6           -3249.
[illegible] 1    3            -940.
[illegible] 0    3            7220.
[illegible] 0    3           14376.
[illegible] 0    3           10761.
[illegible] 0    2           15947.
[illegible] 0    1            5680.
PR        0      0           11108.
[illegible] 0   -1           20000.
PR        0     -2           15843.
PR        1      3           19511.
OO        0      2           17435.
OO        0      1           17310.
OO        0      0           19954.
OO        0     -1           17049.
OO        0     -2           18223.
OO        0     -3           17937.
OO        0     -4           19989.
OO        0     -5           17983.
OO        0     -6           20000.
LL        1      3           20000.

YY        0      2           20000.
MM        0      1           20000.
YY        0      3           20000.
      HH
WW        0      2           20000.
      HH
WW        0      1           20000.
WW        0      0           20000.
WW        0     -1           20000.
WW        0     -2           20000.
WW        0     -3           20000.
YY        0     -4           20000.
YY        0     -5           20000.
LL        0     -6           20000.
LL        0     -7           20000.
LL        0     -8           20000.
LL        0     -9           20000.
MM        0    -10           20000.
LL        0    -11           20000.
LL        0    -12           20000.
LL        0    -13           20000.
LL        0    -14           20000.
LL        0    -15           20000.
LL        0    -16           20000.
MM        0    -17           20000.
WW        0    -18           20000.
      HH
WW        0    -19           20000.
      HH
IY        1    -20           20000.


        WININ      CHIN
BB      0.43106    0.50332
DD      0.43542    0.81781
GG      0.59066    0.76039
PP      0.67561    0.55757
TT      0.63429    0.64569
KK      0.31316    0.69803

FINAL PHONEME STRING IS:

AA PR OO LL
```

Fig. 41. System Output for L1W31 /owl/

```
CODE FOR THIS SAMPLE IS:    SYNTH  FRAENN  L1W33

                                   7669.
                                  13726.
                                   [illegible]
                                  20000.
                                  20000.

                         WINOUT      CHOUT
                         0.83740     0.83726
                         0.[illegible]     0.[illegible]
                         0.[illegible]     0.[illegible]
                         0.[illegible]     0.[illegible]
                         0.83401     0.[illegible]
                         0.81493     0.[illegible]

                                  20000.
                                   [illegible]
```

```
                                  -2226.
                                   [illegible]
                                  -6548.
                                  -4559.
                                  -8134.
                                   [illegible]

     WININ      CHIN
BB   0.25759    0.27530
[illegible]
KK   0.22540    0.[illegible]

FINAL PHONEME STRING IS:
DD UU EE NN
```

Fig. 42. System Output for L1W33 /ran/

```
CODE FOR THIS SAMPLE IS:    SYNTH  WXEETT  L1W46
MM      0        4              20000.
[illegible]
[illegible]
[illegible]
MM      0        0              20000.

                        WINOUT     CHOUT
BB                      0.15343    0.00000
[illegible]             [illegible] 0.00000
[illegible]             [illegible] 0.00000
[illegible]             [illegible] 0.00000
[illegible]             [illegible] 0.00000
XX                      [illegible]41 0.00000

[illegible]                                20000.
[illegible]

        WININ      CHIN
BB      0.[illegible]  0.[illegible]
DD      0.[illegible]  0.[illegible]
GG      0.[illegible]  0.[illegible]
[illegible]
[illegible]
XX      0.67499    0.15511

FINAL PHONEME STRING IS:
WW I?
```

Fig. 43. System Output for L1W46 /wet/

```
CODE FOR THIS SAMPLE IS:    SYNTH  HHWWUHTT  L1W47

WW        0        4        20000.
     HH
LL        0        3        20000.
     LL
YY        0        2        20000.
     HH
WW        0        1        20000.
     LL
OW        0        0        20000.
     HH

                            WINOUT     CHOUT
[illegible]                 0.30536    0.00000
                            0.16585    0.00000
                            0.15523    0.00000
                            0.02[illegible]    0.00000
                            0.00358    0.00000
                            0.53[illegible]    0.00000

YY        0       -1        20000.
     HH
WW        0       -2        20000.
     HH
WW        0       -3        20000.
     LL
UU        0       -4        20000.
     LL
WW        0       -5        20000.
     HH
ZH        0       -6       -15000.
[illegible]   1     [illegible]   20000.
WW        0        2        20000.
     HH
WW        0        3        20000.
     HH
[illegible]   0        2        20000.
[illegible]        [illegible]   20000.

[illegible]   1   [illegible]   11074.
     HH
CW        0   [illegible]   11107.
     HH
[illegible]   1   [illegible]

               WININ      CHIN
RR             0.82[illegible]    0.43[illegible]
[illegible]    0.[illegible]      [illegible]
[illegible]    0.[illegible]      [illegible]
[illegible]    0.15[illegible]    [illegible]
[illegible]    0.[illegible]      [illegible]
[illegible]    0.05124    0.4[illegible]

FINAL PHONEME STRING IS:
HH UU TT
```

Fig. 44. System Output for L1W47 /what/

```
CODE FOR THIS SAMPLE IS:    SYNTH  WWAIRR  L1W48
                                   20000.

                                   20000.
                                   20000.

                                   20000.
                                   20000.

                               WINOUT     CHOUT
                              0.25459   0.17558
                              0.12455   0.14134
                              0.05505   0.05519
                              0.42714   0.05621
                              0.12554   0.14748
                              0.24546   0.07254

                                   20000.
```

Fig. 45. System Output for L1W48 /wire/

1 1
HH

	WININ	CHIN
BB	0.24578	0.34169
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
EE	0.33118	0.45077
[illegible]	[illegible]	[illegible]
KK	0.70270	0.24262

FINAL PHONEME STRING IS:

HH AI ER DD

Fig. 45 (Con't) System Output for L1W48 /wire/

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
CODE FOR THIS SAMPLE IS:    SYNTH  YYLHRRDD  L1W49

IY                4          2413.
[illegible]

        WINOUT      CHOUT
RR     0.12221     0.24212
[illegible]
KK     0.08853     0.24224

IY                           11232.
[illegible]
                             11550.

UH      0         2          13703.
UH      0         1          13311.
[illegible]

        WININ       CHIN
RR     [illegible] [illegible]
[illegible]
KK     0.61724     0.22541

        FINAL PHONEME STRING IS:
IY RR ER RD
```

Fig. 46. System Output for L1W49 /yard/

CODE FOR THIS SAMPLE IS: SYNTH CHEESSTT L2W8

SH	0	4	-11495.
[illegible]	0	3	-15209.
[illegible]	[illegible]	[illegible]	-17654.
[illegible]	[illegible]	[illegible]	-17543.
[illegible]	[illegible]	[illegible]	-13793.

	WINOUT	CHOUT
BB	0.23312	0.10737
[illegible]	0.45765	0.16635
[illegible]	0.31157	0.14380
[illegible]	0.33306	0.12117
[illegible]	0.42214	0.22646
XX	0.24433	0.24405

	WININ	CHIN
BB	0.33767	0.70102
[illegible]	0.67381	[illegible]
[illegible]	0.15772	[illegible]
[illegible]	0.21814	0.56107
[illegible]	0.61779	[illegible]
XX	0.84577	0.64557

FINAL PHONEME STRING IS:

SH II EE SS TT

Fig. 47. System Output for L2W8 /chest/

```
CODE FOR THIS SAMPLE IS:     SYNTH  GGEIVV  L2W16

TH        0        4            7349.
[illegible]        0        3            5507.
[illegible]        0        2            5406.
[illegible]        0        1            3702.
[illegible]        0        0            1160.

                            WINCUT     CHCUT
EE                        0.85755    0.25556
[illegible]               0.75302    0.40567
[illegible]               0.64108    0.85759
[illegible]               0.25563    0.24500
[illegible]               0.60230    0.09096
[illegible]               0.92116    0.28702

[illegible]

                  WININ      CHIN
EE              [illegible]  [illegible]
[illegible]
[illegible]
KK              0.51096      0.45596

FINAL PHONEME STRING IS:
II EI VV
```

Fig. 48. System Output for L2W16 /gave/

```
CODE FOR THIS SAMPLE IS:   SYNTH  HHIITT  L2W18

[illegible]      4        15559.

[illegible]      3         4455.
                           [illegible]
                           [illegible]
                           [illegible]

[illegible]      2        13[illegible].

[illegible]               1151.
                          [illegible]
                 2         9277.

[illegible]               [illegible]
                 6         [illegible]

       WINOUT     CHOUT                          WININ      CHIN
       0.4[illegible]  0.42[illegible]      BB   0.83257    0.34607
       0.75438    0.45821                   [illegible]  [illegible]  [illegible]
       [illegible]  0.7[illegible]          [illegible]  0.62[illegible]  [illegible]
       [illegible]  0.63[illegible]         [illegible]  [illegible]  0.11[illegible]
       [illegible]  0.16[illegible]         [illegible]  [illegible]  [illegible]
       0.46122    0.42556                   KK   0.6[illegible]  [illegible]

                                                 [illegible]
                                                 [illegible]
                                                 [illegible]
                                                 [illegible]
                                                 [illegible]
                                                 [illegible]
                                                 [illegible]
                                                 -15470.
                                                 [illegible]
                                                 -11759.
                                                  0.

[illegible]                3280.
                           [illegible]
                           [illegible]
                          14[illegible].

[illegible]               -6096.

[illegible]               [illegible]
                           [illegible]
                           [illegible]
                           [illegible]
                           [illegible]
                           2108.

FINAL PHONEME STRING IS:
II TT
```

Fig. 49. System Output for L2W18 /hit/

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

CODE FOR THIS SAMPLE IS: SYNTH HHERTT L2W19

[illegible]	[illegible]	[illegible]	14285.
[illegible]	[illegible]	[illegible]	20000.
[illegible]	[illegible]	[illegible]	15594.
[illegible]	[illegible]	[illegible]	20000.
[illegible]	[illegible]	0	20000.

	WINOUT	CHOUT
[illegible]	0.82350	0.75637
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	0.51136
[illegible]	0.93640	0.72072

	WININ	CHIN
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
KK	[illegible]	[illegible]

[illegible]	[illegible]	[illegible]	20000.
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	9203.
[illegible]	[illegible]	[illegible]	20000.
[illegible]	[illegible]	[illegible]	20000.
[illegible]	[illegible]	[illegible]	16425.
[illegible]	[illegible]	[illegible]	20000.

FINAL PHONEME STRING IS:

KK ER RR ER TT

Fig. 50. System Output for L2W19 /hurt/

```
CODE FOR THIS SAMPLE IS:    SYNTH  JJOW  L2W22

                                   174.
                                 -1213.
                                 14915.

                                 20000.
                                 12022.

                                 14936.
                                 20000.

                                 20000.

                    WINOUT       CHOUT

                    WININ        CHIN

FINAL PHONEME STRING IS:
DD HH UU OX
```

Fig. 51. System Output for L2W22 /jaw/

```
CODE FOR THIS SAMPLE IS:    SYNTH  OUKK  L2W29
UH          0       4         6360.
       HH
[illegible] 0       [illegible]  11178.
                              10738.
                              9743.
       HH
[illegible] 0       0         11700.
       HH

                    WINOUT     CHOUT
[illegible]         0.07537    0.40552
                    0.10517    0.34555
                    0.03443    0.36183
                    0.01550    0.48119
                    0.13423    0.15919
                    0.01049    0.35701

OW          0      -1         9140.
       HH
[illegible] 0      -10        11079.
                   -10        9488.
       HH
[illegible] 0      -4         13078.
                   -8         9105.
       HH
OW          0      -6         6724.
       HH
[illegible] 0      -7         12335.
                   -8         [illegible]
```

```
[illegible]                  15305.
                             [illegible]
                             16812.
                             [illegible]
EE  [illegible]  -1          -953.
OO.                          -1088.
ZH        1       3          0.

      WININ      CHIN
EE    0.60344    0.38871
OO    0.[illegible]  0.[illegible]
OO    0.[illegible]  0.[illegible]
EE    0.66715    0.10347
[illegible]  0.[illegible]  0.[illegible]
KK    0.31795    0.13253

FINAL PHONEME STRING IS:
OU
```

Fig. 52. System Output for L2W29 /oak/

```
CODE FOR THIS SAMPLE IS:     PPYYOO   L2W34

NN              0           4               14280.
          HH
TH              0           3                9842.
TH              0           2                8850.
RR              0          -1               17588.
IY              0           3               17325.
IY              0           2               17351.
IY              0          -1                7675.
IY              0           3                4024.
UU              0           3                4655.
UU              0           3                7322.
UU              0           3                8023.
UU              0          -1               21730.
UU              0           3                2896.
UU              0           2                9425.
RR              0          -1               20000.
RR              0           0               15228.


                                 WINOUT     CHOUT
BB                              0.73568    0.34617
DD                              0.77349    0.72748
GG                              0.75660    0.56119
PP                              0.22946    0.18161
TT                              0.65916    0.85716
KK                              0.45031    0.59858


RR              0          -1               19016.
RR              1           3               19287.
DD              0           2               20000.
DD              0          -1               17296.
DD              0           0               17343.
DD              0          -1               18255.
DD              0          -1               18450.
DD              0           3               18526.
DD              0          -1               18113.
DD              0           4               18234.
DD              0          -1               18423.
DD              0           3               20000.
DD              1           3               20000.
DD              0           2               20000.
DD              0          -1               20000.
DD              0           3               20000.
WH              0           3               20000.
YY              0           2               20000.
          HH
ZH              1           3                   0.


      WINMIN     CHIN
BB   0.35120   0.50268
DD   0.29470   0.40559
GG   0.35700   0.00000
PP   0.70000   0.61578
TT   0.03792  -0.37041
KK   0.35672   0.00000


 FINAL PHONEME STRING IS:
TT UU CC
```

Fig. 53. System Output for L2W34 /pew/

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

CODE FOR THIS SAMPLE IS: RROOMMSS L2W35

RR	0	4	9828.
RR	0	3	14935.
UU	0	3	18992.
UU	0	3	20000.
UU	0	2	20000.
MM	0	1	20000.
YY	0	0	20000.

	WINOUT	CHOUT
BB	0.73106	0.78262
DD	0.87294	0.80024
GG	0.76491	0.79828
PP	0.58814	0.60457
TT	0.81552	0.79601
KK	0.76446	0.79316

YY	0	-1	20000.
LL	0	-2	20000.

HH

MM	0	-3	20000.
MM	0	-4	20000.
LL	0	-5	20000.
LL	0	-6	20000.
MM	0	-7	20000.
UU	0	-8	20000.
OO	0	-9	20000.
OO	0	-10	20000.
OO	0	-11	10264.
HH	1	3	9263.
HH	0	2	17643.
OO	0	1	16788.
OO	0	0	17900.
OO	0	-1	19497.
OO	0	-2	17271.
OO	0	-3	18942.
OO	0	-4	17781.
OO	0	-5	17079.
OO	0	-6	16936.
OO	0	-7	17581.
OO	0	-8	18931.

OO	0	-9	15472.
OO	0	-10	14642.
RR	1	3	9997.
NN	0	3	2251.
NN	0	3	-1363.
NN	0	3	1284.
NN	0	3	1831.
NN	0	2	96.
NN	0	1	131.
NN	0	0	413.
NN	0	-1	-1110.
NN	0	-2	-1290.
NN	0	-3	-2231.
NN	0	-4	-1149.
NN	0	-5	1415.
ER	0	-6	15000.
RR	0	-7	20000.
UU	0	-8	15570.
IY	0	-9	2702.
SS	1	3	-18083.
SS	0	3	-18654.
SS	0	2	-16405.
SS	0	1	-17438.
SS	0	0	-18919.
SS	0	-1	-19329.
SS	0	-2	-21057.
SS	0	-3	-21033.
SS	0	-4	-13944.
SS	1	3	-2590.

	WININ	CHIN
BB	0.40119	0.00000
DD	0.68554	0.00000
GG	0.64531	0.00000
PP	0.47323	0.00000
TT	0.84250	-0.02130
KK	0.69326	0.00000

FINAL PHONEME STRING IS:

DD MM OO NN SS

Fig. 54. System Output for L2W35 /rooms/

```
CODE FOR THIS SAMPLE IS:   SYNTH  SHOU  L2W37

                     WINOUT     CHOUT

                     WININ      CHIN

FINAL PHONEME STRING IS:
SH UU OU
```

Fig. 55. System Output for L2W37 /show/

CODE FOR THIS SAMPLE IS: SYNTH SSMMAARRTT L2W38

[illegible]	0	4	-19343.
[illegible]	[illegible]	3	[illegible]
[illegible]	[illegible]	2	[illegible]
[illegible]	[illegible]	1	-19573.
[illegible]	[illegible]	0	-19377.

	WINOUT	CHOUT
[illegible]	0.02256	0.02822
[illegible]	0.21443	[illegible]
[illegible]	[illegible]	0.02577
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	0.01064	0.02782

[illegible]	0	-1	-17451.
[illegible]	[illegible]	-2	[illegible]
[illegible]	[illegible]	-3	[illegible]
[illegible]	[illegible]	-4	[illegible]
[illegible]	[illegible]	3	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	-2	[illegible]
[illegible]	[illegible]	-3	-430.
[illegible]	[illegible]	-4	17377.
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	12314.

Fig. 56. System Output for L2W38 /smart/

	WMIN	CMIN
BB	0.45771	0.55519
[illegible]	0.[illegible]	0.[illegible]
[illegible]	0.[illegible]	0.[illegible]
[illegible]	0.[illegible]	0.[illegible]
[illegible]	0.[illegible]	0.[illegible]
ZK	0.[illegible]47	0.[illegible]

FINAL PHONEME STRING IS:

SS MM AA RR II TT

Fig. 56. (Con't) System Output for L2W38 /smart/

```
CODE FOR THIS SAMPLE IS:   SYNTH  TEAETT  L2W41

NG         0         4                -3248.
      HH
TH         0         3                 9001.
[illegible]          2                22000.
[illegible]          1                13[illegible].
[illegible]          0                 4[illegible]5.

                            WINOUT       CHOUT
BB                          2.64[illegible]6   0.31159
[illegible]                 0.[illegible]   0.[illegible]
[illegible]                 0.[illegible]   0.[illegible]
[illegible]                 0.97[illegible]44   0.[illegible]
[illegible]                 0.2[illegible]04   0.[illegible]
KK                          0.64550      0.43[illegible]

                            WININ        CHIN
BB                          0.91[illegible]   0.[illegible]
[illegible]                 0.[illegible]   0.[illegible]
[illegible]                 0.[illegible]   0.[illegible]
[illegible]                 0.[illegible]   0.[illegible]
[illegible]                 [illegible]   [illegible]
KK                          0.[illegible]   0.[illegible]

15727.
[illegible]
4971.

FINAL PHONEME STRING IS:
TH EE TT
```

Fig. 57. System Output for L2W41 /that/

```
CODE FOR THIS SAMPLE IS:      TEEENN   L2W42

NN          0         4            9636.
       HH
OO          0         3           20000.
       HH
TE          0         2           17334.
TH          0         1           13418.
PP          0         0           11545.


                             WINOUT     CHOUT
BB                          0.86918    0.10006
DD                          0.00000    0.24938
GG                          0.85954    0.00000
PP                          0.73620    0.14011
TT                          0.29676    0.00000
KK                          0.52598    0.06195


TE          0        -1           15021.
PP          0        -2           14456.
TH          1         3            9817.
TH          0         2            6391.
TE          0         3            7354.
EE          0         3            3932.
EE          0         2            3038.
TT          0         1            3592.
       HH
EE          0         3            -336.
EE          0         2           -1384.
EE          0         1           -1501.
EE          0         0            -730.
EE          0        -1           -1899.
EE          0        -2             665.
EE          0        -3           -1645.
FF          0        -4           -4700.
       HH
TH          1         3           -6817.
       HH
TH          0         2           -6251.
       HH


NN          0         1           -3720.
NN          0         3           -4025.
NN          0         2           -4193.
NN          0         1           -3027.
NN          0         0           -1291.
NN          0        -1            -933.
NN          1         3           -1009.
NN          0         2            1309.
NN          0         1            3049.
NN          0         3            4095.
NN          0         3            5028.
NN          0         3           14137.
NN          0         2           14250.
       HH
ZH          1         3               0.


     WININ      CHIN
BB  0.24955   0.00000
DD  0.11655   0.18723
GG  0.35500   0.00000
PP  0.24975   0.10232
TT  0.10507  -0.06071
KK  0.37982   0.19281


FINAL PHONEME STRING IS:
EE NN
```

Fig. 58. System Output for L2W42 /then/

```
CODE FOR THIS SAMPLE IS:   SYNTH  WWEI  L2W46

        WINOUT     CHOUT
        [illegible] [illegible]

        WININ      CHIN
        [illegible] [illegible]

        FINAL PHONEME STRING IS:

MM II IY
```

Fig. 59. System Output for L2W46 /way/

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
CODE FOR THIS SAMPLE IS:    SYNTH WWIITH  L2W48

WW     0      4        20000.
   HH
[illegible]            20000.
[illegible]

          WINOUT     CHOUT
[illegible] 0.03163   0.02020
            0.01102   0.00000
            0.03461   0.00000
            0.03008   0.00040
            0.00000   0.00000
            0.04342   0.00000

[illegible]            20000.
[illegible]            1576.

[illegible]            -13          [illegible]
[illegible]            -10          20000.
   HH
NG        0             2            3272.
   HH
[illegible] 0           3            337.
   HH
ZZ        0             3            2793.
   HH
ZZ        0             2            -344.
   HH
[illegible] 0           1            4527.
[illegible] 0           0            3333.
   HH
[illegible]

        WININ     CHIN
[illegible] 0.01148   0.32722
[illegible]
ZZ      0.00000   0.05555

FINAL PHONEME STRING IS:
WW II HH
```

Fig. 60. System Output for L2W48 /with/

```
CODE FOR THIS SAMPLE IS:    SYNTH  YYUHNNGG  L2W49

IY          0          4              3540.
IY          0          3              7258.
IY          0          2              6342.
IY          0          1              7050.
IY          0          0              7275.

                            WINOUT     CHOUT
BB                        0.16238   0.16238
[illegible]               0.37214   0.21569
[illegible]               0.24640   0.11004
[illegible]               0.21448   0.11281
[illegible]               0.24437   0.13838
XX                        0.25328   0.21249

IY          0         -1             14002.
[illegible]                          12795.
[illegible]                          12833.
IY                       1           11154.
       HH
[illegible] 0          3             10579.
       HH
[illegible]                           7035.
[illegible]                          14112.
[illegible]                          12616.
[illegible]                           8573.
[illegible]                          15125.
[illegible]                          11120.
[illegible]                           7772.
[illegible]                          14035.
[illegible]                          14140.
[illegible]                          11344.
[illegible]                          14283.
       HH
[illegible]              -5          11737.
AA                       -6           9032.
       HH
AA          3            -7           7306.

UH          [illegible]  -8          11937.
       [illegible]
[illegible] 0            -9          10942.
       HH
[illegible] 0           -10          15570.
[illegible]             -11          [illegible]
[illegible]                          [illegible]
                                      7351.
                                      3071.
[illegible]
       HH
[illegible] 1             3          [illegible]

           WININ      CHIN
BB       0.31055    0.32565
[illegible]
[illegible]
[illegible]
[illegible]
XX       0.32073    0.25325

     FINAL PHONEME STRING IS:

IY TH RR NN GG
```

Fig. 61. System Output for L2W49 /young/

Bibliography

1. Coker, C. H. "Model of the Mouth Used for Synthesis Calculations." IEEE Soundings, 1: Tape Recording (August 1971).

2. Flanagan, J. L. Speech Analysis, Synthesis and Perception. New York: Springer-Verlac, 1965.

3. House, A. "On Vowel Durations in English." Journal of the Acoustical Society of America, 33: 1174-1178 (December 1961).

4. Itakura, F. "Minimum Prediction Residual Principle Applied to Speech Recognition." IEEE Acoustics, Speech and Signal Processing, ASSP-23: 67-72 (February 1975).

5. Kempelen, W. v. Le Mechanisme de la Parole, suivi de la Description d'une Machine Parlante. Vienna: J. V. Degen, 1791.

6. Klatt, D. H. and K. N. Stevens. "On the Automatic Recognition of Continuous Speech: Implications from a Spectrogram-Reading Experiment." IEEE Audio and Electroacoustics, AV-21: 210-217 (June 1973).

7. Koenig, W., et al. "The Sound Spectrograph." Journal of the Acoustic Society of America, 17: 19-49 (July 1946).

8. Mattingly, I. G. Synthesis by Rule of American English. Unpublished Dissertation. New Haven, Conn.: Yale University, June 1968.

9. Mundie, J. R., et al. "Signal Processing Principles Revealed by an Auditory System Model." Proceedings of the 5th Congress of the Deutsche Gesellschaft für Kybernetik: 292-307 (March 1973).

10. Mundie, J. R., et al. The CxC System: General Description. Wright-Patterson Air Force Base, Ohio: Aerospace Medical Division, In Processing.

11. Neiderjohn, R. J. "A Mathematical Formulation and Comparison of Zero-Crossing Analysis Techniques WhichHave Been Applied to Automatic Speech Recognition." IEEE Acoustics, Speech and Signal Processing, ASSP-23: 373-379 (August 1975).

12. Newel, A., et al. "Speech Understanding Systems: Final Report of a Study Group." Amsterdam: North-Holland/American Elsevier, 1973.

13. Pierce, J. R. "Whither Speech Recognition?" Journal of the Acoustical Society of America, 46: 1049-1051 (December 1969).

14. Rabiner, L. R. Speech Synthesis by Rule: An Acoustic Domain Approach. Unpublished Dissertation. Cambridge, Mass.: Massachussetts Institute of Technology, June 1967.

15. Steer, R. W., Jr., et al. Design of an Active COC Filter for Audio Frequency Signal Processing. AMRL TR 75-78. Wright-Patterson Air Force Base, Ohio: Aerospace Medical Division, January 1976.

16. Turn, R., et al. Military Applications of Speech Understanding. ARPA 189-1. Washington, D. C.: Advanced Research Projects Administration, June 1974.

17. Warmuth, D. B. Speech Synthesis by a Programmable Digital Filter. GE/EE/75-41. Wright-patterson Air Force Base, Ohio: Air Force Institute of Technology, December 1975.

18. White, G. M. and R. B. Neely. "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering and Dynamic Programming." IEEE Acoustics, Speech and Signal Processing, ASSP-24: April 1976.

19. White, G. M. "Speech Recognition: A Tutorial Overview." IEEE Computer, 9: 40-53 (May 1976).

Appendix A

Synthesis Strategy

In this appendix the method of taking the phonemic representation of what is to be "said" and converting it to an acceptable form for the Model 4516 synthesizer is presented. The representation of the desired utterance includes phonemes, stress marks (ST), word boundaries (blanks), pauses (period, question mark or comma) and a terminal symbol (END). The inputs to the Model 4516 are the 24 parameters introduced in Chapter III which include 10 pole or zero frequencies and their associated bandwidths, two volume controls, a pitch period duration, and a threshold for voiced fricatives. Some of the rules and methods which follow were derived directly from Rabiner (Ref. 14). His work is the starting point from which we began.

Phoneme Characteristics

Each phoneme has a unique steady state characterization. This characterization includes the first three formant target frequencies (F_1,F_2,F_3), the voiced amplitude (A_V), noise amplitude (A_N), a frequency range ($\Delta 1$,$\Delta 2$,$\Delta 3$) around each of the formant targets, and a duration. The formant targets for stops and fricatives are virtual targets; the formants do not actually reach the targets because voicing is replaced by voiceless sound (fricatives) or amplitude drops (stops). The frequency ranges are used to determine when a transition to a new phoneme is complete. When all three formants have moved to within hertz of the specified target and the additional duration requirements are met, the transition is defined to be complete. In the case of a nasal or

fricative the characterization must also include the frequency and bandwidth of the nasal pole and zero or the fricative pole and zero. Table V on page 127 presents the formant targets, amplitudes, frequency ranges, and the additional durations of the various phonemes. The diphthongs, affricatives and aspirant are not included for reasons which will be discussed later.

Formant Motion

In connected speech the speaker moves from one phoneme to another in a continuous manner. Although some phonemes are known to influence others two or three removed, the principle effects are produced by the adjacent phonemes. Transitions, in this strategy, are a function of only the two adjacent phonemes. Transitions, as one might expect, are smooth and continuous. We have adopted Rabiner's strategy of using a critically damped second degree differential equation to control motion in the frequency space. He chose a second degree equation because it provided a good fit to the observed data. He made it critically damped because only a single time constant is necessary to completely characterize the response to a forcing function. Since he found this algorithm worked very well, we have chosen to follow this procedure. The equation is

$$x(t)=Af+(Ai-Af)\exp(-t/\tau)+\left[Vi+\frac{(Ai-Af)}{\tau}\right]t\,\exp(-t/\tau) \qquad (3)$$

where

$x(t)$=formant value as a function of time

τ=time constant in ms

Ai=formant target of previous phoneme

TABLE V

PHONEME CHARACTERISTICS

Phoneme	F_1	F_2	F_3	A_V	A_N	A1	A2	A3	DURATION
IY	270	2290	3010	100	0	40	40	110	50
II	390	1990	2550	88	0	50	50	90	20
EE	530	1840	2480	60	0	50	55	90	20
AE	660	1720	2410	45	0	40	40	75	50
UH	580	1190	2390	40	0	50	50	50	20
AA	730	1090	2442	38	0	25	40	80	50
OW	570	840	2410	28	0	40	40	80	50
UU	440	1020	2240	43	0	50	50	65	30
OO	350	1300	3900	85	0	40	45	55	50
ER	450	1275	1700	47	0	30	20	30	50
BB	150	600	3000	20	0	50	75	120	20
PP	150	800	1750	0	0	50	40	80	20
MM	280	900	2200	120	0	17	17	40	30
DD	440	1300	1700	20	0	50	50	160	20
TT	440	2200	3000	0	0	50	30	100	10
NN	280	1300	2000	120	0	17	17	100	30
GG	220	1300	1450	20	0	50	50	100	20
KK	220	1300	3300	0	0	50	30	70	20
NG	280	1700	2600	120	0	17	17	100	50
FF	175	900	2400	0	50	20	34	80	20
VV	175	1100	2400	65	35	10	15	40	20
TH	200	1400	2200	0	99	20	28	68	00
TE	200	1600	2200	50	90	10	15	100	00
SS	200	1300	2500	0	40	20	28	50	50
ZZ	200	1300	2500	50	90	20	30	50	20
SH	175	1800	2050	0	99	10	34	100	50
ZH	175	1800	2000	50	40	10	40	100	20
WW	300	610	2200	45	0	25	40	150	00
LL	380	1000	2575	75	0	25	80	150	30
RR	420	1300	1600	50	0	40	80	100	30
YY	300	2200	3065	58	0	25	110	200	00
RO	295	845	1315	80	0	30	80	100	00

Af=formant target of current phoneme

Vi=velocity of the formant at $t=t_0$

For computer simulation of this method, the above equation was Z-transformed using impulse invariant technique (to preserve the time response to an impulse) to obtain the following difference equation

$$x(nT)=2kx(nT-T)-k^2x(nT-2T)+(1-k)^2F(nT-T) \qquad (4)$$

where

T=sampling time (PER)

x(nT)=formant position at time nT

$k=e^{-T/\tau}$

τ=time constant in ms

and

F(nT)=formant target at time nT

Formant data is used to define intrinsic phoneme durations and is the basic mechanism from which all timing is controlled.

Time Constants

Each formant may move from one target to the next at different rate; thus, a time constant is necessary for each formant in the transition. In this implementation there are 31 basic phonemes which gives 961 possible combinations. Since three formants are controlled for each phoneme there are 2883 possible time constants. However, by using certain approximations and phoneme groupings the number of time constants was reduced to a more workable 371.

Formant Changes

All three formants may not begin motion toward their new targets simultaneously. In three cases, vowel-stop, vowel-nasal, and consonant-vowel the initiation of the transition of the first formant is delayed by $\tau 2-\tau 1$ ms where $\tau 2$ is the time constant for formant two and $\tau 1$ is the time constant of formant one. This delay serves to emphasize the transition of formants two and three which are significant for proper perception in these cases.

Nasals

The nasal pole and zero and the bandwidth of formant one are shifted so that they are in position when the amplitude is switched for the nasal and are returned to a nominal value at the end of the nasal. These shifts take about 50 ms. If the nasal is preceded by a voiced sound, the shift of these values can be heard in the voiced branch of the synthesizer and this branch is being excited. This effect is not undesirable because the slight nasalization of the preceding sound is found in natural speech.

Fricatives

The fricative pole and zero move in the same manner as the nasal pole and zero but the movement is not normally heard. If the sound preceding a voiceless fricative is voiced, the pitch of the last 40 ms of the phoneme is reduced slightly. This is a clue that a voiceless fricative is coming and is understandable on a physical basis because the vocal cords are stopping. In a voiceless fricative the formant targets are virtual targets and are used only for controlling the

transition to and from a voiced sound and for timing. They are not excited during the fricative.

In a voiced fricative, on the other hand, the formants are excited and when the amplitude of the output of the formant one pole exceeds a given threshold the fricative branch is enabled. The output of the two branches are summed and each pitch-period of the output of the synthesizer looks like a dampened sine-wave with noise added above a certain amplitude.

Stops

All stops are characterized by a rapid shut-down of the volume of the preceding phoneme and a period of silence of about 100 ms. The release of the stop, however, is determined by whether the stop is voiced or voiceless. The voiced stop has a rapid release of voicing of the following phoneme and a slight overshoot ($\sim$20%) of the volume. A voiceless stop has a short-duration burst of fricative noise and a period of aspiration ($\sim$40 ms) followed by the onset of voicing. The amplitude of the voicing is rapidly increased to the value of the succeeding phoneme.

When a vowel is the first sound in an utterance, the speaker performs a "glottal stop." That is, a rapid onset of voicing very similar to the release of a voiced stop. For example, the word /ate/ in initial position differs from /gate/, /bait/, or /date/ only in that the point of release is the glottis. This synthesis scheme incorporates the glottal stop.

Aspirant H and Whispering

In American English the aspirant H is always followed by a vowel or W. In this simulation H is generated by forming and lengthening the succeeding sound and aspirating the first part of it. Although the duration of the aspiration is context dependent, an average value of 100 ms was used. Aspiration and whispering are accomplished by driving the voiced path with the noise source (negative A_V) for the voiced (positive A_V) sounds and are easily accomplished in this simulation.

Diphthongs and Affricatives

The diphthongs EI, OU, AI, OI, and AU are treated as two vowel sequences, namely EE→II for EI, OW→OO for OU, AA→II for AI, OW→II for OI, and AA→OO for AU. The input diphthongs are automatically replaced in the input string with the appropriate two vowels and the length of each vowel is reduced by 20 ms.

The affricates, CH and J, have a low frequency of occurrence in American speech. CH appears only 0.44% of the time and J appears 0.52% of the time (Ref. 2 :5). CH has a stop gap of silence followed by a burst of noise, similar to a T, and has a long period of noise, similar to SH, following the burst. J has a voiced release similar to a D followed by a long period of voiced frication similar to ZH. These two sounds are simulated by treating them as the two phoneme sequences TTSH and DDZH.

Amplitude Changes

Amplitude and source characteristics must be tuned to each other and to the formant transitions or essential cues will not be present in

the signal. Amplitude changes should begin after the formants begin moving toward the new phonemes but well before the movement is complete. The amount of delay greatly affects the amount of transition that is heard and, therefore, the recognizability of the phoneme. For example, in a stop-vowel transition, if the amplitude change is turned on too late or too slowly the transition is lost and the stop will not be correctly perceived. In a vowel-stop transition, on the other hand, if the amplitude is reduced too quickly the transition is likewise lost, thus making it difficult to recognize the stop. In all cases requiring a change in source characteristics the time of source change is based on the time and rate of transition of formant one.

For consonant-vowel transitions a large percentage of the formant transitions should occur after the source characteristics are changed. Thus, in this implementation, the switch takes place $\tau 1$ ms (about 30% of the transition has taken place) after transition begins. For vowel-consonant transitions again $\tau 1$ ms is used except when the consonant is a stop. In that case the source characteristic is changed $1.5\tau 1$ ms (about 45%) after formant transition begins. $1.5\tau 1$ ms is used in consonant-consonant transitions when a stop is in the second position; otherwise the delay is $\tau 1$. When a change in source from voiced to voiceless is required, this change is initiated at the same time as the amplitude change is begun.

In general, the rate at which the source characteristics change is determined by the next phoneme. The only exception is a transition from a stop. In this case the change is controlled by the stop rather than the phoneme which follows it.

Pitch

Without pitch modulation rules the synthesizer speaks in a flat monotone. The color of natural speech results in a large part from the pitch inflection we impose on the basic speech message reflecting our feelings about the context. Pitch variation is not essential to speech synthesis, but the monotony of a monotone detracts from the quality of the synthetic speech. Therefore we devised some simple rules to produce pitch variations.

During a voiced sound the pitch is varied inversely with the value of formant one over a range of 133 to 118 Hz as formant one varies between 270 to 730 Hz. The pitch may also be altered during the last 200 ms of an utterance. For a statement the pitch drops by 45 Hz in this period; whereas, for a question, the pitch rises by 45 Hz. The pitch may be held steady at the end of an utterance when desired, such as for the recitation of a list of words.

Stress

When a vowel is stressed in natural speech three things happen; the pitch rises, the amplitude increases, and the phoneme is lengthened. All three effects have been incorporated in this simulation. The pitch and amplitude are both raised by aout 20% for the duration of the vowel. The amount the vowel is lengthened is a function of the following phoneme. In a study by House (Ref. 3) it was found that the longest stressed vowels are followed by voiced sounds. House's experiments were for isolated words and the results were found to be unacceptable for connected speech. Therefore, following Rabiner's lead, all durations

were reduced by 100 ms. When a diphthong is stressed the effect depends on the phoneme. For EI and OU the second sound is stressed; whereas, for AI, OI, and AU the first sound is stressed.

Contextual Effects

The context in which a phoneme appears may have an effect on the way it is uttered. The areas of contextual effects which were considered are word boundaries, initiation and shut-down, and certain phoneme combinations.

Word Boundaries. Word boundaries have only a minimal effect on connected speech. In most cases the speaker rolls right over them as if they did not exist. There is only one effect that was addressed. An R in word initial position is changed to the allophone RO.

Initiations and Shut-Downs. When a phoneme is at the beginning or end of an utterance it is handled differently than if it is internal to the utterance. An initial vowel has a glottal stop for a beginning followed by about 50 ms of steady state. An initial stop is, of course, preceded by a period of silence so the stop begins with the release. All other phonemes begin with the formants at steady state and last for approximately 50 ms.

A final vowel is lengthened by about 25%. A final vowel, fricative, or nasal has a gradual shut-down of source amplitude. In a voiced fricative the voice source shuts off more rapidly than the noise source and thus they sound like their fricative counterparts for the last 50 ms of the utterance. A final stop has a low level, short /UH/ inserted before the source is shut off.

When a period or comma is encountered the utterance is terminated as above and a pause, or period of silence, is generated ($\sim$100 ms for comma and $\sim$150 ms for period). After the pause a new utterance is initiated.

Phoneme Combinations. When the back vowels, OW, U, OO, are succeeded by P, F, TH, S, B, M, V, TE, or Z the second formant of the second phoneme is increased by 400 Hz.

Appendix B

Automatic Synthetic Speech Recognition Programs

Main Program (ROSSRE)

```
      IMPLICIT INTEGER (A-Z)
      COMMON STATS(460).DAT(256).IEND
      REAL TALLY.MSRMOM(4.25).AMPRAT.HHMOM(4.25)
      REAL TODAY(3).NOW(2)
      DIMENSION CHANEL(32).CODE(50)
      BYTE FILSPC(10).FILE3(10).FILERE(10).FILEHH(10)
      DATA TODAY/3*'    '/
      DATA FILSPC/'T'.'S'.'T'.'D'.'A'.'T'.'.'.'D'.'I'.'S'/
      DATA FILERE/'R'.'E'.'A'.'L'.'S'.'P'.'.'.'D'.'I'.'S'/
      DATA FILEHH/'R'.'E'.'A'.'L'.'H'.'H'.'.'.'D'.'I'.'S'/
      STFLAG=0
      ICNT=10
      TALLY=0.0
      IEND=0

C     THIS IS A PROGRAM TO RECOGNIZE SPEECH FROM DATA PRODUCED
C     BY C X C .  THE DATA IS STORED IN A FILE CALLED 'TSTDAT.DIS'.
C     THE PROGRAM CAN WORK ON EITHER OF TWO SETS OF PROTOTYPES.
C     THE PROGRAM IS CALLED ROSSRE.

      WRITE (6.2060)
 2060 FORMAT('   WHAT TYPE OF SPEECH?'/.
     2'   R::=REAL'/'   CR::=SYNTHETIC'/)
      READ (5.2070) SP
 2070 FORMAT(A1)
      IF (SP.EQ.'R') CALL ASSIGN(7.FILERE.ICNT)
      IF (SP.EQ.'P') CALL ASSIGN(3.FILEHH.ICNT)
      DEFINE FILE 3(11.256.U.IVAR3)
      DEFINE FILE 7(44.256.U.IVAR2)

C     READ MASTER SET OF MOMENTS

      READ (7'26) DAT
      DO 10 J=1.25
      JJ=(J-1)*4
      DO 10 K=1.4
   10 MSRMOM(K.J)=DAT(JJ+K)*10.0
      DO 20 J=1.11
      JJ=(J-1)*4+100
      DO 20 K=1.4
   20 HHMOM(K.J)=DAT(JJ+K)*10.0
      CALL ASSIGN (1.FILSPC.ICNT)
      DEFINE FILE 1(300.256.U.IVAR)
      IBLKNO=3
      READ (1'IBLKNO)DAT
      WRITE (6.2050)
 2050 FORMAT('   ENTER PRINT TERMINAL'/'   B::= BOTH LINE'.
     2' PRINTER AND CONSOLE'/'   CR::= CONSOLE ONLY'/)
      READ (6.2055) CON
      IF (CON.EQ.' ') GO TO 1
 2055 FORMAT(A1)
      WRITE (6.2010)
 2010 FORMAT(' ENTER FILE NAME'/)
      READ (6.2015) FILE3
 2015 FORMAT(10A1)
      CALL ASSIGN(2.FILE3.ICNT)
      WRITE (6.2020)
 2020 FORMAT('    ENTER CODE FOR THIS SAMPLE'/)
      READ (5.2030) CODE
 2030 FORMAT(50A1)
      CALL DATE (TODAY)
```

```
      CALL TIME (NOW)
      WRITE (2,2040) CODE,TODAY,NOW
2040  FORMAT('   CODE FOR THIS SAMPLE IS:  ',50A1,' ON '3A4' AT '2A4/)
C     FIND FIRST PP MARKER.  SEARCH FOR UP TO 6 MS.
1     DO 4 J=2,254,2
      IF(DAT(J-1).EQ.-1) STOP
      TEMP=DAT(J)/32
      CH=DAT(J)-TEMP*32
      I=J
      IF(CH.EQ.0) GO TO 5
      TALLY=TALLY+DAT(J-1)/2
      IF(TALLY.LT.600.0) GO TO 4
      I=0
      READ(1'3) DAT
      GO TO 5
4     CONTINUE
      IBLKNO=DAT(255)+1
      READ(1'IBLKNO) DAT
      GO TO 1
5     AMPRAT=2000.0
C     GO TO WORK
6     CALL ICONPE(I,TALLY,CHANEL,TOT,STFLAG,AMPRAT)
      CALL COMPRE(MSEMOM,CHANEL,TOT,STFLAG,AMPRAT,HHMOM,CON)
      IF(IEND.EQ.1) STOP
      GO TO 6
      END



      SUBROUTINE ICONPE(I,TALLY,CHANEL,TOT,STFLAG,AMPRAT)
      IMPLICIT INTEGER (A-Z)
      REAL TOTAL,RTEMP,TALLY,MTRX(32),TIMEX
      REAL AMPRAT,LASAMP
      COMMON STATS(480),DAT(256),IEND
      DIMENSION CH(2),CHANEL(32)
      DATA LASAMP/0.0/
      DATA MTRX,TIMEX/33*0.0/
C     THIS PROGRAM CONVERTS DATA IN TSTDAT.DIS TO HISTOGRAM
C     AND CHANNEL FIRINGS.
5     DO 10 J=1,32
10    CHANEL(J)=0
      DO 11 J=1,480
11    STATS(J)=0
      TALLY=0.0
      TOTAL=0.0
12    I=I+2
      IF(I.LT.256) GO TO 14
      IBLKNO=DAT(256)+1
      READ (1'IBLKNO)DAT
      I=2
14    IF(DAT(I-1).NE.-1) GO TO 16
      IEND=1
      GO TO 20
C     DATA IS PACKED INTO GROUPS OF TWO WORDS.  FIRST WORD IS
C     THE TIME (IN 5 MICRO-SEC INTERVALS) SINCE LAST PULSE.
C     SECOND WORD IS, FROM LOW ORDER BIT, CHANNEL (5 BITS),
C     FLAGS (3 BITS), SECOND CHANNEL (5 BITS), AND MORE FLAGS.
C     CHANNEL IN LOW ORDER BYTE IS LOW ORDER CHANNEL (0-31).
C     SECOND CHANNEL IS ZERO IF NOT USED.
C     UNPACK THE DATA
16    TEMP=DAT(I)/32
      CH(1)=DAT(I)-(TEMP*32)
      TEMP=TEMP/8
      TEMP2=TEMP/32
      CH(2)=TEMP-(TEMP2*32)
      TIMEX=DAT(I-1)/2.0+TIMEX
```

```
C       LOOK FOR SILENCE FOR STOPS.

        IF(DAT(I-1)/2.GT.2500) STFLAG=1
        TALLY=TALLY+DAT(I-1)/2.0
        IF(TALLY.GT.1000.0) GO TO 20

C       LOOK FOR PP MARKER OR ELAPSED TIME OF 10 MS.

        IF(CH(1).EQ.0) GO TO 20
        DO 18 J=1,2
        CHAN=CH(J)+1
        IF(CHAN.NE.2) GO TO 165
        AMPRAT=TIMEX-LASAMP
        LASAMP=TIMEX

C       IGNORE CHANNELS ZERO AND ONE - ENTER OTHERS IN HISTOGRAM.

 165    IF(CHAN.LE.2)GO TO 18
        IF(MTRX(CHAN).EQ.0.0) GO TO 17
        L=TIMEX-MTRX(CHAN)
        IF (L.GT.480.OR.L.LT.1) GO TO 17
        STATS(L)=STATS(L)+1
        TOTAL=TOTAL+1.0
 17     MTRX(CHAN)=TIMEX
        CHANEL(CHAN)=CHANEL(CHAN)+1
 18     CONTINUE
        GO TO 12
 20     IF(IEND.EQ.0) GO TO 26
        WRITE(6,1050)
 1050   FORMAT('    END OF DATA REACHED')
 26     IF(TOTAL.LE.1.0.AND.IEND.LE.0) GO TO 5
        IF(TOTAL.LE.1.0) TOTAL=1.0

C       NORMALIZE HISTOGRAM TO 300 POINTS.

        TOT=TOTAL
        TOTAL=300.0/TOTAL
        DO 28 J=1,480
        RTEMP=STATS(J)*TOTAL
        STATS(J)=RTEMP
        IF(RTEMP-STATS(J).GE.0.5) STATS(J)=STATS(J)+1
 28     CONTINUE
        RETURN
        END



        SUBROUTINE COMPRE(MSRMOM,CHANEL,TOT,STFLAG,AMPRAT,HMMOM,CON)
        IMPLICIT INTEGER (A-Z)
        REAL RAWMOM,MOM(3),LOW,HIGH,MSRMOM(4,25),ERR(25),ERR1,
       2MATCH(25),SORT,CHERR(25),REGION,AMPRAT
        REAL HMMOM(4,25),HERR(25),HMATCH(25),HCHERR(25),
       2HVAL,VAL
        COMMON STATS(480),DAT(256),IEND
        DIMENSION STP(3),OFSET(3),MSORT(25),PSORT(25),CSORT(25)
        DIMENSION CHANEL(32),HMSORT(25),HCSORT(25),HPSORT(25)
        DATA STP/54,90,210/
        DATA PRCNT/5/
        DATA OFSET/0,90,120/
        DATA HERR/25*1000.0/
        DATA HMATCH,HCHERR/50*0.0/

C       THIS PROGRAM COMPUTES THE MOMENTS OF THE INCOMING SIGNAL
C       SEGMENT AND COMPARES ITS MOMENTS, HISTOGRAM, AND CHANNEL
C       FIRINGS TO THE MASTER PROTOTYPES.

C       CALCULATE THE MOMENTS.

        RAWMOM=0.0
        DO 5 K=1,100
 5      RAWMOM=RAWMOM+(101-K)*STATS(K)
```

```
      DO 10 K=1.230
10    RAWMOM=RAWMOM+K*STATS(K+100)
      TEMMOM=0
      DO 15 K=231.480
15    TEMMOM=TEMMOM+STATS(K)
      RAWMOM=RAWMOM+230.0*TEMMOM
      DO 20 J=1.3
        MOM(J)=0.0
        EN=STP(J)
        DO 20 K=1.EN
          KK=K+OFSET(J)
20    MOM(J)=MOM(J)+STATS(KK)*K
      MOM(3)=MOM(3)+210.0*TEMMOM
      IF(RAWMOM.GT.20000.0) RAWMOM=20000.0

C     CALCULATE DISTANCE OF INCOMING SIGNAL MOMENTS FROM MASTERS.

      REGION=100000.0
      LOW=RAWMOM-REGION
      HIGH=RAWMOM+REGION
      DO 30 J=1.25
        ERR(J)=10000.0
        IF(J.EQ.1.OR.J.EQ.6.OR.J.EQ.9) GO TO 22
        IF(MSRMOM(1.J).LT.LOW.OR.MSRMOM(1.J).GT.HIGH) GO TO 30
22      ERR1=0.0
        DO 25 K=2.4
25      ERR1=ERR1+(MSRMOM(K.J)-MOM(K-1))**2
        ERR(J)=SQRT(ERR1)
30    CONTINUE

C     CALCULATE DISTANCE OF INCOMING SIGNAL MOMENTS FROM HH.

      DO 36 J=1.11
      HERR(J)=10000.0
      IF(HHMOM(1.J).LT.LOW.OR.HHMOM(1.J).GT.HIGH) GO TO 36
      ERR1=0.0
      DO 33 K=2.4
33    ERR1=ERR1+(HHMOM(K.J)-MOM(K-1))**2
      HERR(J)=SQRT(ERR1)
36    CONTINUE

C     RANK MOMENT RESULTS.

      TYPE=1
      CALL RANK(ERR,MSORT,TYPE)
      HFLAG=0
      CALL MATCHR(MSRMOM.HIGH.LOW.MATCH.PSORT.CHERR.CHANEL.
     2CSORT.HFLAG)

C     RANK HH MOMENT RESULTS.

      HFLAG=1
      CALL RANK(HERR.HMSORT.TYPE)

C     CALL SUBROUTINE TO CALCULATE DISTANCE OF HISTOGRAMS AND
C     CHANNEL FIRINGS.

      CALL MATCHR(HHMOM.HIGH.LOW.HMATCH.HPSORT.HCHERR.CHANEL.
     2HCSORT.HFLAG)

C     CALL PARTITIONING ALGORITHM.

      CALL PARTIT(AMPRAT.RAWMOM.MATCH.PARFLG.PARCNT.STFLAG)
      TOTVAL=300
      BEST=25

C     RANK RESULTS OF THREE METHODS.

      DO 40 J=1.25
      RANKS=PSORT(J)+CSORT(J)+MSORT(J)
      IF(RANKS.GE.TOTVAL) GO TO 40
      TOTVAL=RANKS
      BEST=J
40    CONTINUE
      TOTVAL=200
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
C       RANK RESULTS OF THREE METHODS FOR HH.

        DO 50 J=1.11
        RANKS=HPSORT(J)+HCSORT(J)+HMSORT(J)
        IF(RANKS.GE.TOTVAL) GO TO 50
        TOTVAL=RANKS
        HBEST=J
   50   CONTINUE

C       TEST IF HH IS MOST PROBABLE.  IF SO NEGATE HBEST AS
C       AN INDICATOR.

        HVAL=(1.0-HMATCH(HBEST))*(1.0-HCHERR(HBEST))*HERR(HBEST)
        VAL=(1.0-MATCH(BEST))*(1.0-CHERR(BEST))*ERR(BEST)
        IF(HVAL.LT.VAL) HBEST=-HBEST

C       CALL SUBROUTINE TO RECOGNIZE PHONEMES.

        CALL IREC(STFLAG.CHANEL.BEST.PARFLG.PARCNT.HBEST.
       2RAWMOM.CON)
        RETURN
        END




        SUBROUTINE IREC(STFLAG.CHANEL.BEST.PARFLG.PARCNT.HBEST.
       2RAWMOM.CON)
        IMPLICIT INTEGER (A-Z)
        COMMON STATE(480).DAT(255).IEND [illegible]
        REAL TOPWIN(12.6).TOPCH(12.6).WINMAT(12.6).CHMAT(12.6).
       2WINNOR(12.6).CHNOR(12.6).RECST(25).RECCNT.HRECST
        REAL RAWMOM
        DATA WINNOR.CHNOR/144*0.0/
        DATA RECST.HRECST/26*0.0/
        DATA RECST/25*0.0/
        DATA REC/100*26/
        DATA RECNT/0/
        DATA TOPWIN.TOPCH/144*0.0/
        DATA SOUND/'IY'.'II'.'EE'.'AE'.'UH'.'AA'.'OW'.'UU'.'OO'.'ER'
       1'MM'.'NN'.'NG'.'WW'.'LL'.'RR'.'YY'.
       2'FF'.'SS'.'SH'.'TH'.'VV'.'TZ'.'ZZ'.'ZH'.'  '.'BB'.
       3'DD'.'GG'.'PP'.'TT'.'KK'.'CH'.'JJ'.'HH'.
       4'EI'.'AI'.'OI'.'OU'.'AU'/
        DATA INIT/1/

C       THIS PROGRAM RECOGNIZES PHONEMES. CALCULATES STOP
C       NORMALIZATIONS. AND PRINTS RESULTS.

C       CALL ALGORITHM FOR STOP FUNCTIONS.

 1000   FORMAT(3XA3.2I10.F17.0)
        CALL WINDPE(WINMAT.CHMAT.CHANEL)
        STREC=0

C       RECORD HIGHEST STOP CORRELATIONS.


        DO 5 J=1.12
        DO 5 K=1.6
        IF(TOPWIN(J.K).LT.WINMAT(J.K)) TOPWIN(J.K)=WINMAT(J.K)
        IF(TOPCH(J.K).LT.CHMAT(J.K))TOPCH(J.K)=CHMAT(J.K)
    5   CONTINUE
   10   WRITE (6.1000)SOUND(BEST).
       2PARFLG.PARCNT.RAWMOM
        IF(HBEST.LT.0) WRITE (6.2000) SOUND(35)
        IF(CON.EQ.' ') GO TO 999
        WRITE (2.1000) SOUND(BEST).PARFLG.PARCNT.RAWMOM
        IF(HBEST.LT.0) WRITE (2.2000) SOUND(35)
 2000   FORMAT(10XA4)
```

```
C       UPDATE RECOGNITION MATRIX

  999   RECST(BEST)=RECST(BEST)+0.01
        IF(HBEST.LT.0) HRECST=HRECST+0.01

C       NORMALIZE FINAL PORTION STOP PARAMETERS AND CALL
C       STOP DETERMINATION ALGORITHM WHEN NECESSARY.

   30   IF(PARCNT.NE.0.AND.IEND.EQ.0) GO TO 36
        DO 32 J=2,12,2
        DO 32 K=1,6
        CHNOR(J,K)=CHMAT(J,K)
   32   WINNOR(J,K)=WINMAT(J,K)
        IF(STFLAG.EQ.0.AND.INIT.EQ.0.AND.IEND.EQ.0) GO TO 36
        DO 34 J=1,12
        DO 34 K=1,6
        TOPWIN(J,K)=(TOPWIN(J,K)-WINNOR(J,K))/(1.0-WINNOR(J,K))
   34   TOPCH(J,K)=(TOPCH(J,K)-CHNOR(J,K))/(1.0-CHNOR(J,K))
        CALL STDET(TOPWIN,TOPCH,STBEST,INIT,CON)
        STFLAG=0
        IF(STBEST.GT.50) GO TO 36
        IF(IEND.EQ.1) STREC=1
        RECPT=RECPT+1
        REC(RECPT)=STBEST+26

C       UPDATE INITIAL PORTION OF STOP PARAMETERS WHEN NECESSARY.

   36   IF(PARCNT.NE.0) GO TO 40
        INIT=0
        DO 37 K=1,6
        DO 37 J=1,11,2
        CHNOR(J,K)=CHMAT(J,K)
   37   WINNOR(J,K)=WINMAT(J,K)
        DO 39 K=1,6
        DO 39 J=1,12
        TOPWIN(J,K)=0.0
   39   TOPCH(J,K)=0.0

C       IF PARTITION FLAG IS SET, RECOGNIZE PHONEME.

   40   IF(PARFLG.LE.0) GO TO 70

C       IF MOST LIKELY PHONEME WAS RECOGNIZED FOR LESS THAN THREE
C       SEGMENTS - IGNORE IT.

        IF(STREC.EQ.1) GO TO 70
        RECCNT=HRECST
        IF(IEND.EQ.1) RECCNT=0.0
        DO 42 J=1,25
        IF(RECCNT.LT.RECST(J)) RECCNT=RECST(J)
   42   CONTINUE
        IF(RECCNT.LT.0.03) GO TO 50
        TYPE=-1

C       RANK CANDIDATES

        CALL RANK(RECST,RANKED,TYPE)
        DO 44 J=1,25
        IF(RANKED(J).NE.1) GO TO 44
        BEST=J
        GO TO 46
   44   CONTINUE
   46   IF(HRECST.LT.RECST(BEST)) GO TO 475
        HALTR=BEST
        BEST=35
  475   CHG=0

C       CALL POST PROCESSOR

        CALL POSTPR(BEST,REC,RECPT,CHG,HALTR)
        IF(CHG.GT.0) GO TO 50
        RECPT=RECPT+1
        REC(RECPT)=BEST
```

```
C        ZEPO RECOGNITION MATRIX.
  60     DO 65 J=1.25
  65     RECST(J)=0.0
         HRECST=0.0
  70     IF(IEND.EQ.0) RETURN
         WRITE (6.2500)
2500     FORMAT(//'     FINAL PHONEME STRING IS:'/)
         IF (COM.NE.' ') WRITE (2.2500)
         DO 80 J=1.RECPT.20
         KK=J+19
         WRITE(6.3000)(SOUND(REC(K)).K=J.KK)
         IF(COM.NE.' ') WRITE(2.3000)(SOUND(REC(K)).K=J.KK)
3000     FORMAT(20A3/)
  80     CONTINUE
         STOP
         END


         SUBROUTINE MATCHR(MSRMOM.HIGH.LOW.ERR.PSORT.
        2ERR2.CHANEL.CSORT.HFLAG)
         IMPLICIT INTEGER (A-Z)
         REAL LOW.HIGH.MSRMOM(4.25).MOM.ERR(25).TEMP.SQRT.STATSQ
         REAL ERR2(25).CHANSQ
         COMMON STATS(480).DAT(256).IEND
         DIMENSION PSORT(25).DATUM(256).CSORT(25).CHANEL(32)

C        THIS PROGRAM MATCHES INCOMING SIGNAL HISTOGRAMS AND
C        CHANNEL FIRINGS AGAINST MASTERS FOR STEADY STATE
C        SOUNDS AND HH AS APPROPRIATE.

         STATSQ=0.0
         DO 5 J=1.480
         IF(STATS(J).GT.100) STATS(J)=100
   5     STATSQ=STATSQ+STATS(J)**2
         IF(STATSQ.LE.1.0) STATSQ=1.0
         CHANSQ=0.0
         DO 7 J=1.32
   7     CHANSQ=CHANSQ+CHANEL(J)**2
         IF(CHANSQ.LE.1.0) CHANSQ=1.0
         DO 30 J=1.25
           ERR(J)=0

          IF(HFLAG.EQ.1.AND.J.GT.11) GO TO 30
           MOM=MSRMOM(1.J)
           IF(MOM.LT.LOW.OR.MOM.GT.HIGH) GO TO 30
           IF(HFLAG.EQ.0) READ (7'J)DATUM
           IF(HFLAG.EQ.1) READ (3'J) DATUM
           PLC=0

C         MASTER HISTOGRAMS ARE PACKED.  FROM LOW ORDER BIT -
C         FIRST 6 BITS ARE VALUE IN MATRIX. REST OF BITS ARE SUB-
C         SCRIPT FOR THAT VALUE.

            DO 10 K=1.224
              SUB=DATUM(K)/64
              IF(SUB.LT.PLC) GO TO 20
              VAL=DATUM(K)-(SUB*64)
              ERR(J)=ERR(J)+(VAL*STATS(SUB))
  10        PLC=SUB
  20        CONTINUE
            TEMP=STATSQ*DATUM(226)
          IF(TEMP.LT.1.0) TEMP=1.0
           ERR(J)=ERR(J)/SQRT(TEMP)
          ERR2(J)=0.0
          DO 25 K=1.32
  25      ERR2(J)=ERR2(J)+(CHANEL(K)*DATUM(224+K))
          TEMP=CHANSQ*DATUM(225)*10.0
          IF(TEMP.LT.1.0) TEMP=1.0
          ERR2(J)=ERR2(J)/SQRT(TEMP)
  30      CONTINUE
          TYPE=-1
```

```
C       RANK CANDIDATES FOR HISTOGRAM AND CHANNEL FIRING RESULTS.

        CALL RANK(EPR.PSORT.TYPE)
        CALL RANK(EPR2.CSORT.TYPE)
        RETURN
        END




        SUBROUTINE WINDRE(WINMAT.PATMAT.CHANEL)
        IMPLICIT INTEGER (A-Z)
        REAL WINSQ.MATCH(31).CHEPR(31).SQRT
        REAL MATCH2(6).CHEPR2(6).CHANSQ.RTEMP
        REAL PATMAT(12.6).WINMAT(12.6)
        COMMON STATS(480).DAT(256).IEND
        DIMENSION CHANEL(32).WIN(14).STRT(14).STP(14).DATUM(256)
        DATA STRT/1.20.48.64.90.110.150.183.211.230.260.300.375.434/
        DATA STP/30,48.74,95,120.155.185.213,240,270,310.380.437,480/
C       THIS PROGRAM CALCULATES WINDOW FUNCTIONS AND CHANNEL FIRINGS
C       OF INCOMING SIGNAL FOR STOP DETECTION AND IDENTIFICATION.

C       CALCULATE WINDOW FUNCTIONS.

        DO 20 J=1,14
        WIN(J)=0
        BEG=STRT(J)
        ST=STP(J)
        DO 20 K=BEG.ST
   20   WIN(J)=WIN(J)+STATS(K)
        WINSQ=0.0
        DO 30 J=1.14
        RTEMP=WIN(J)
   30   WINSQ=WINSQ+RTEMP*RTEMP
        DO 35 J=1,12
        DO 35 K=1.6
   35   WINMAT(J.K)=0.0

C       MATCH WINDOW FUNCTIONS AGAINST VARIOUS STOP MASTERS.

        DO 50 M=1.6
        MM=25+M
        READ (7'MM) DATUM
        DO 50 L=1.2
        KK=(M-1)*2+L
        OO=0
        IF(L.GT.1) OO=90
        DO 50 J=1.6
        JJ=(J-1)*15+OO
        DO 40 K=1.14
   40   WINMAT(KK.J)=WINMAT(KK.J)+WIN(K)*DATUM(JJ+K)
        RTEMP=DATUM(JJ+15)
        IF(RTEMP.LT.0.0) RTEMP=-RTEMP*10.0
        RTEMP=WINSQ*RTEMP
        IF(RTEMP.LT.1.0) RTEMP=1.0
   50   WINMAT(KK.J)=WINMAT(KK.J)/SQRT(RTEMP)

C       CALCULATE CHANNEL FIRINGS.

        CHANSQ=0.0
        DO 60 J=3.22
        RTEMP=CHANEL(J)
   60   CHANSQ=CHANSQ+RTEMP*RTEMP
        IF(CHANSQ.LE.1.0) CHANSQ=1.0

C       MATCH CHANNEL FIRINGS AGAINST VARIOUS STOP MASTERS.

        DO 100 L=1.12
        MM=32+L
        READ (7'MM) DATUM
```

```
      DO 85 J=1.6
      PATMAT(L.J)=0.0
      DD=(J-1)*32
      DO 80 K=3.32
      RTEMP=CHANEL(K)
 80   PATMAT(L.J)=PATMAT(L.J)+DATUM(DD+K)*RTEMP
      RTEMP=DATUM(DD+1)
      IF(RTEMP.LT.0.0) RTEMP=-RTEMP*10.0
      RTEMP=CHANSQ*RTEMP
      IF(RTEMP.LE.1.0) RTEMP=1.0
 85   PATMAT(L.J)=PATMAT(L.J)/SQRT(RTEMP)
100   CONTINUE
      RETURN
      END




      SUBROUTINE POSTPR(BEST.REC.RECPT.CHG.HALTR)
      IMPLICIT INTEGER (A-Z)
      COMMON STATS(480).DAT(256).IEND
      DIMENSION REC(100)

C     THIS PROGRAM POST PROCESSES THE PHONEME STRING.

C     FINAL HH NOTT ALLOWED.

      IF(BEST.EQ.35.AND.IEND.EQ.1) BEST=HALTR
      IF(RECPT.LT.1) RETURN

C     IF PREVIOUS PHONEME WAS AN HH, CHECK IF IT IS VALID.

      IF(REC(RECPT).NE.35) GO TO 5
      IF(BEST.GT.10) GO TO 3
      REC(RECPT)=35
      RETURN
  3   IF(BEST.NE.14) GO TO 4
      REC(RECPT)=35
      RETURN
  4   REC(RECPT)=HALTR

C     IGNORE REPEATED PHONEMES.

  5   IF(BEST.NE.REC(RECPT)) GO TO 10
      RECPT=RECPT-1
      RETURN

C     TEST FOR JJ.

 10   IF(BEST.NE.25.OR.REC(RECPT).NE.28) GO TO 20
      REC(RECPT)=34
      CHG=1
      RETURN

C     TEST FOR CH.

 20   IF(BEST.NE.20.OR.REC(RECPT).NE.31) GO TO 30
      REC(RECPT)=32
      CHG=1
      RETURN

C     TEST FOR REPEATED SECOND PHONEME OF DIPHTHONGS.

 30   IF(BEST.GT.2) GO TO 70
      IF(REC(RECPT).LT.36.OR.REC(RECPT).GT.38) GO TO 40
      CHG=1
      RETURN
 40   IF(REC(RECPT).NE.3) GO TO 50
      REC(RECPT)=25
      CHG=1
      RETURN
```

```
 50    IF(REC(RECPT).NE.6) GO TO 60
       REC(RECPT)=37
       CHG=1
       RETURN
 60    IF(REC(RECPT).NE.7) RETURN
       REC(RECPT)=38
       CHG=1
       RETURN
 70    IF(BEST.NE.9) GO TO 100
       IF(REC(RECPT).LT.39) GO TO 80
       CHG=1
       RETURN
C      TEST FOR DIPHTHONGS.
 80    IF(REC(RECPT).NE.7) GO TO 90
       REC(RECPT)=39
       CHG=1
       RETURN
 90    IF(REC(RECPT).NE.6) RETURN
       REC(RECPT)=40
       CHG=1
       RETURN
100    IF(BEST.NE.19.OR.REC(RECPT).NE.24) RETURN
       CHG=1
       RETURN
       END


       SUBROUTINE PARTIT(AMPRAT,RAWMOM,MATCH,PARFLG,PARCNT,
      2STFLAG)
       IMPLICIT INTEGER (A-Z)
       REAL RAWMOM,MATCH(25),SQRT,AMPAVE,MOMAVE,CORAVE(3)
       REAL AMPNOR,MOMNOR,CORNOR(3)
       REAL AMPRAT,ABS,AMPL,CORSS
       REAL AMPLAS(3),MOMLAS(3),CORLAS(3,3),CORCHG(3),AMPCHG,MOMCHG
       COMMON STATS(480),DAT(256),IEND
       DIMENSION FLAGS(3)
       DATA INIT/2/
       DATA AMPAVE,MOMAVE,CORAVE/5*0.0/
       DATA AVEPT,FLAGS/4*0/
       DATA MOMLAS,AMPLAS,CORLAS/15*0.0/
C      THIS PROGRAM DETERMINES PARTITION BOUNDARIES BASED ON
C      CORRELATIONS AGAINST IY, AA, AND OO; AMPLITUDE OF THE
C      INCOMING SIGNAL; AND TTHE RAW MOMENT.  FLAG( ) IS THE
C      THREE FLAGS FOR THESE PARTITION PARAMETERS.
       DO 10 J=1,3
 10    FLAGS(J)=FLAGS(J)-1
       AMPL=AMPRAT/100.0
C      INITIALIZE ON FIRST TTWO SEGMENTS.
       IF(INIT.LT.1) GO TO 30
       DO 20 K=1,3
       AMPLAS(K)=AMPL
       MOMLAS(K)=RAWMOM
       CORLAS(1,K)=MATCH(1)
       CORLAS(2,K)=MATCH(6)
 20    CORLAS(3,K)=MATCH(9)
       AMPNOR=AMPL
       MOMNOR=RAWMOM
       CORNOR(1)=MATCH(1)
       CORNOR(2)=MATCH(6)
       CORNOR(3)=MATCH(9)
 30    INIT=INIT-1
```

```
C       AVERAGE PARTITION PARAMETERS OVER LAST THREE SEGMENTS.

        AVEPT=AVEPT+1
        IF(AVEPT.GT.3) AVEPT=1
        AMPLAS(AVEPT)=AMPL
        MOMLAS(AVEPT)=PAWMOM
        CORLAS(1,AVEPT)=MATCH(1)
        CORLAS(2,AVEPT)=MATCH(6)
        CORLAS(3,AVEPT)=MATCH(9)
        AMPAVE=0.0
        MOMAVE=0.0
        DO 32 J=1,3
  32    CORAVE(J)=0.0
        DO 34 J=1,3
        AMPAVE=AMPAVE+AMPLAS(J)/3.0
        MOMAVE=MOMAVE-MOMLAS(J)/3.0
        DO 34 K=1,3
  34    CORAVE(K)=CORAVE(K)+CORLAS(K,J)/3.0

C       CALCULATE CHANGES IN PARTITION PARAMETERS.

        AMPCHG=AMPAVE-AMPNOR
        MOMCHG=MOMAVE-MOMNOR
        DO 36 J=1,3
  36    CORCHG(J)=CORAVE(J)-CORNOR(J)
        CORSS=0.0
        DO 40 K=1,3
  40    CORSS=CORSS+CORCHG(K)**2
        IF(CORSS.LE.0.0) GO TO 50
        CORSS=SQRT(CORSS)

C       TEST TO SEE IF ANY PARAMETERS HAVE EXCEDED THEIR THRESHOLD.
C       IF SO, SET APPROPRIATE FLAGS AND CHANGE APPROPRIATE
C       BASELINES.
  50    IF(CORSS.GT.0.1750) FLAGS(1)=3
        IF(ABS(AMPCHG).GT.1.0) FLAGS(2)=3
        IF(ABS(MOMCHG).GT.3000.0) FLAGS(3)=3
        PAR=0

C       TEST FOR PARTITION BOUNDARY.

        DO 60 J=1,3
        IF(FLAGS(J).GT.0) PAR=PAR+1
  60    CONTINUE
        PARFLG=0
        TEMP=PAR
        IF(PARCNT.GT.0) PAR=0
        IF(TEMP.GE.2) PARCNT=4
        IF(PAR.GE.2) PARFLG=1
        IF(IEND.EQ.1) PARFLG=1
        PARCNT=PARCNT-1
        IF(FLAGS(2).EQ.3) AMPNOR=AMPAVE
        IF(FLAGS(3).EQ.3) MOMNOR=MOMAVE
        IF(FLAGS(1).NE.3) GO TO 80
        DO 70 J=1,3
  70    CORNOR(J)=CORAVE(J)

C       RESET PARAMETER FLAGS AT BOUNDARY.

  80    IF(PARCNT.NE.3) RETURN
        DO 90 J=1,3
  90    FLAGS(J)=0
        RETURN
        END
```

```
      SUBROUTINE STDET(TOPWIN,TOPCH,STBEST,INIT,CON)
      IMPLICIT INTEGER (A-Z)
      COMMON STATS(460),DAT(255),IEND
      REAL TOPWIN(12,6),TOPCH(12,6),VAL(2)
      DIMENSION BEST(2),SOUND(6)
      DATA SOUND/'BB','DD','GG','PP','TT','KK'/

C     THIS PROGRAM DETECTS INITIAL AND FINAL STOPS AND
C     IDENTIFIES ALL STOPS.

      CALL TOPS(TOPWIN)
      CALL TOPS(TOPCH)
      WRITE (6,500)
      IF(CON.NE.' ') WRITE (2,400)
 400  FORMAT(///)
      IF(CON.NE.' ') WRITE (2,500)
 500  FORMAT(10X'WININ'5X'CHIN'6X'WINOUT'5X'CHOUT')
      DO 5 J=1,6
      WRITE (6,1000)SOUND(J),TOPWIN(1,J),TOPCH(1,J),TOPWIN(2,J),
     2TOPCH(2,J)
      IF(CON.NE.' ') WRITE(2,1000)SOUND(J),TOPWIN(1,J),TOPCH(1,J)
     2TOPWIN(2,J),TOPCH(2,J)
1000  FORMAT(A5,4F10.5)
   5  CONTINUE
      IF (CON.NE.' ') WRITE (2,400)

C     DETECT INITIAL OR FINAL STOP IF SUM OF APPROPRIATE
C     CORRELATIONS EXCEEDS 1.5

      DO 7 J=1,2
      DO 7 K=1,6
   7  TOPWIN(J,K)=TOPWIN(J,K)+TOPCH(J,K)
      IF (INIT.LE.0.AND.IEND.LE.0) GO TO 25
      JJ=2
      IF(IEND.GT.0) JJ=1
      DO 10 J=1,6
      IF(TOPWIN(JJ,J).GE.1.5) GO TO 25
  10  CONTINUE
      STBEST=99
      RETURN

C     FIND STOP WITH HIGHEST CORRELATION SUM

  25  VAL(1)=-1.0
      VAL(2)=-1.0
      DO 30 J=1,2
      DO 30 K=1,6
      IF(TOPWIN(J,K).LE.VAL(J)) GO TO 30
      VAL(J)=TOPWIN(J,K)
      BEST(J)=K
  30  CONTINUE

C     IDENTIFY STOP.

      IF(INIT.LE.0) GO TO 40
      STBEST=BEST(2)
      RETURN
  40  IF(IEND.LE.0) GO TO 50
      STBEST=BEST(1)
      RETURN
  50  JJ=BEST(1)+3
      IF(JJ.GT.6) JJ=BEST(1)-3
      IF(BEST(1).NE.BEST(2).AND.BEST(2).NE.JJ) GO TO 60
      STBEST=BEST(2)
      RETURN
  60  STBEST=BEST(2)
      RETURN
      END
```

```
      SUBROUTINE RANK(ERR,ISORT,ITYPE)
      INTEGER STATS,DAT
      DIMENSION ERR(25),ISORT(25),TERR(25)
      COMMON STATS(480),DAT(256),IEND

C     THIS PROGRAM RANKS 25 VALUES IN DECREASING ORDER.

      DO 10 J=1,25
      TERR(J)=ERR(J)
      IF(ITYPE.LT.0) TERR(J)=1-ERR(J)
 10   CONTINUE
      DO 50 J=1,25
      VAL=100000.0
      DO 30 K=1,25
      IF(TERR(K).GT.VAL) GO TO 30
      IPLC=K
      VAL=TERR(K)
 30   CONTINUE
      ISORT(IPLC)=J
 50   TERR(IPLC)=1000000.0
      RETURN
      END
```

```
      SUBROUTINE IOFSET (BEG,IBLKNO)
      IMPLICIT INTEGER (A-Z)
      COMMON STATS(480),DAT(256),IEND
      REAL TALLY,RTEMP
      TALLY=0.0
      WRITE (6,1000)
 1000 FORMAT('  HOW FAR (IN MS) DO YOU WANT BEGIN OFFSET?'/)
      READ (6,1010) OFFSET
 1010 FORMAT(I3)
      RTEMP=OFFSET*100.0
 2    DO 4 J=1,253,2
      IF(DAT(J).EQ.-1) GO TO 10
      TALLY=TALLY+DAT(J)/2.0
      BEG=J+1
      IF(TALLY.GE.RTEMP) RETURN
 4    CONTINUE
      IBLKNO=DAT(255)+1
      READ(1'IBLKNO) DAT
      GO TO 2
 10   IEND=1
      WRITE (6,1020)
 1020 FORMAT('  END OF DATA REACHED BEFORE OFFSET.'/)
      RETURN
      END
```

```
      SUBROUTINE TOPS(TOP)
      INTEGER STATS,DAT
      COMMON STATS(480),DAT(256),IEND
      DIMENSION TOP(12,6)

C     THIS PROGRAM DETERMINES THE CLOSEST MATCH FOR EACH STOP
C     AGAINST INCOMING SIGNAL.

      DO 10 L=1,2
      DO 10 J=1,6
      DO 10 K=2,6
      KK=(K-1)*2+L
      IF(TOP(KK,J).GT.TOP(L,J))TOP(L,J)=TOP(KK,J)
 10   CONTINUE
      RETURN
      END
```

Program for Storing Steady-State Prototypes (ROSSST)

```
      IMPLICIT INTEGER (A-Z)
      REAL SCALE(12),TOTAL,RTEMP,TODAY(3),NOW(2),CHANSQ
      REAL RAWMOM,MOM(3),MTRX(32),TALLY,TIME2,TIMEX,EMTRX(32)
      COMMON STATS(480),DAT(256),IEND
      DIMENSION CHANEL(32),CH(2),IMOM(4),STP(3),OFSET(3)
      DIMENSION PRNT(120),CODE(30)
      DIMENSION PATTRN(480),SOUND(31)
      BYTE FILSPC(10),FILERE(10),FILEHH(10)
      DATA MTRX/32*0.0/
      DATA STP/54,90,210/
      DATA OFSET/0,50,120/
      DATA FILERE/'R','E','A','L','S','P','.','D','I','S'/
      DATA FILEHH/'R','E','A','L','H','H','.','D','I','S'/
      DATA TODAY/3*'    '/
      DATA FILSPC/'T','S','T','D','A','T','.','D','I','S'/
      DATA SOUND/'IY','II','EE','AE','UH','AA','OW','UU','OO','ER',
     1'MM','NN','NG','WW','LL','RR','RO',
     2'FF','SS','SH','TH','VV','TE','ZZ','ZH',
     3'BB','DD','GG','PP','TT','KK'/

C     THIS IS ROSSST.  IT STORES MASTER PATTERNS (STEADY STATE
C     AND HH) FOR USE WITH OTHER ROSS__ PROGRAMS.

      ICNT=10
      IEND=0
 2020 FORMAT('    ENTER LABEL FOR THIS SAMPLE'/)
 2030 FORMAT(30A2)
 2040 FORMAT('     LABEL FOR THIS SAMPLE IS:  ',30A2' ON '3A4,
     2' AT '2A4//)
      WRITE(6,2050)
 2050 FORMAT('   WHAT TYPE OF SPEECH?'/'     R::=REAL'/,
     2'   CR::=SYNTHETIC'/)
      READ (6,2060) SP
 2060 FORMAT(A1)
      IF(SP.EQ.'R')CALL ASSIGN(7,FILERE,ICNT)
      IF(SP.EQ.'R')CALL ASSIGN(3,FILEHH,ICNT)
      CALL ASSIGN (1,FILSPC,ICNT)
      DEFINE FILE 1(200,255,U,IVAR)
      DEFINE FILE 3(11,255,U,IVAR4)
      DEFINE FILE 7(36,255,U,IVAR2)
      IBLKNO=3
      READ (1'IBLKNO)DAT
      TALLY=0.0

C     CALL ALGORITHM TO FIND BEGINNING OF DATA TO BE STORED
C     AS A MASTER PATTERN.

      CALL SOFSET(BEG,IBLKNO)
      IF(IEND.EQ.1) STOP

C     FILL MTRX FOR 10 MS TO BEGIN DATA.

    2 DO 4 J=BEG,254,2
      I=J
      IF(DAT(J-1).EQ.-1) GO TO 24
      TEMP=DAT(J)/32
      CH(1)=DAT(J)-(TEMP*32)
      TEMP=TEMP/8
      TEMP2=TEMP/32
      CH(2)=TEMP-(TEMP2*32)
      TALLY=TALLY+DAT(I-1)/2.0
      MTRX(CH(1)+1)=TALLY
      MTRX(CH(2)+1)=TALLY
      IF(TALLY.GE.1000.0) GO TO 6
    4 CONTINUE
      IBLKNO=DAT(256)+1
      READ(1'IBLKNO)DAT
      BEG=2
      GO TO 2
```

```
 6      BEG=I
        BEG1=I
        IPTBLK=IBLKNO
        TIMEX=TALLY
        TIME2=TALLY
        TALLY=0.0
        DO 60 J=1,32
C       SEARCH FOR FP MARKER FOR UP TO 20 MS.
 60     EMTRX(J)=MTRX(J)
 62     DO 64 J=BEG,254,2
        IF (DAT(I-1).EQ.-1) GO TO 24
        TEMP=DAT(J)/32
        CH(1)=DAT(J)-TEMP*32
        TEMP=TEMP/8
        TEMP2=TEMP/32
        CH(2)=TEMP-TEMP2*32
        I=J
        TALLY=TALLY+DAT(I-1)/2.0
        TIMEX=TIMEX+DAT(I-1)/2.0
        MTRX(CH(1)+1)=TIMEX
        MTRX(CH(2)+1)=TIMEX
        IF(CH(1).EQ.0) GO TO 66
        IF(TALLY.LT.2000.0) GO TO 64
        I=BEG1
        READ(1'IPTBLK) DAT
        TIMEX=TIME2
        DO 63 K=1,32
 63     MTRX(K)=EMTRX(J)
        GO TO 66
 64     CONTINUE
        IBLKNO=DAT(255)+1
        READ(1'IBLKNO) DAT
        BEG=2
        GO TO 62
 66     TOTAL=0.0
        DO 10 J=1,32
 10     CHANEL(J)=0
        DO 11 J=1,480
 11     STATS(J)=0
        TALLY=0.0
C       UNPACK DATA FROM TSTDAT.DIS AND CONVERT IT TO HISTOGRAM
C       AND CHANNEL FIRINGS.
 12     I=I+2
        IF(I.LT.256) GO TO 14
        IBLKNO=DAT(255)+1
        READ (1'IBLKNO)DAT
        I=2
 14     IF(DAT(I-1).NE.-1) GO TO 16
        IEND=1
        GO TO 31
 16     TEMP=DAT(I)/32
        CH(1)=DAT(I)-(TEMP*32)
        TEMP=TEMP/8
        TEMP2=TEMP/32
        CH(2)=TEMP-(TEMP2*32)
        TIMEX=TIMEX+(DAT(I-1)/2.0)
        TALLY=TALLY+(DAT(I-1)/2.0)
        IF(TALLY.GT.1000.0) GO TO 31
        IF(CH(1).EQ.0) GO TO 31
        DO 18 J=1,2
        CHAN=CH(J)+1
        IF(CHAN.LT.2)GO TO 18
        IF(MTRX(CHAN).EQ.0.0) GO TO 17
        L=TIMEX-MTRX(CHAN)
        IF(L.GT.480.OR.L.LT.1) GO TO 17
        STATS(L)=STATS(L)+1
        TOTAL=TOTAL+1.0
 17     MTRX(CHAN)=TIMEX
        CHANEL(CHAN)=CHANEL(CHAN)+1
 18     CONTINUE
        GO TO 12
```

```
   31   IF(IEND.EQ.0) GO TO 26
   24   WRITE(5,1050)
 1050   FORMAT('    END OF DATA REACHED')
   26   K=1
        WRITE (5,1055)TOTAL
 1055   FORMAT('    TOTAL='F5.0/)

C       NORMALIZE HISTOGRAM TO 300 POINTS.

        TOTAL=300.0/TOTAL
        DO 200 J=1,480
        PATTRN(J)=0
        RTEMP=STATS(J)*TOTAL
        STATS(J)=RTEMP
        IF(RTEMP-STATS(J).GE.0.5) STATS(J)=STATS(J)+1
        IF(STATS(J).LE.63) GO TO 190
        STATS(J)=63
        WRITE (6,5000)
 5000   FORMAT ('  STATS EXCEEDED 63'/)
  190   IF(STATS(J).LE.0) GO TO 200

C       PACK HISTOGRAM FOR STORAGE.

        PATTRN(K)=J*64
        PATTRN(K)=PATTRN(K)+STATS(J)
        K=K+1
  200   CONTINUE
        STATSQ=0
        DO 40 J=1,480
   40   STATSQ=STATSQ+STATS(J)**2
        CHANEL(1)=0
        CHANSQ=0.0
        DO 42 J=3,32
   42   CHANSQ=CHANSQ+CHANEL(J)*CHANEL(J)
        CHANEL(1)=CHANSQ/10.0
        WRITE (6,990) STATSQ,CHANEL(1)
  990   FORMAT('    SUM OF SQUARES IS 'I10,
       2'    SUM OF CHANNEL SQUARES IS 'I10)

C       PRINT HISTOGRAM IF DESIRED.

        WRITE (5,1000)
 1000   FORMAT('    STATISTICS PRINTED?'/)
 1010   FORMAT(1A1)
        READ (6,1010)PR
        IF(PR.NE.'Y') GO TO 201
        WRITE (6,2020)
        READ (6,2030) CODE
        CALL DATE(TODAY)
        CALL TIME(NOW)
        WRITE (5,2040) CODE,TODAY,NOW
        DO 55 J=1,4
        DO 45 K=1,120
   45   PRNT(K)=' '
        DO 52 K=1,16
        DO 50 L=1,120
        IF(STATS(120*(J-1)+L).GE.16-(K-1)) PRNT(L)='!'
   50   CONTINUE
        WRITE (5,2000) PRNT
 2000   FORMAT(2X,120A1)
   52   CONTINUE
        WRITE(5,2002)
 2002   FORMAT(2X,120('-')/2X,12(9X,'I'))
        DO 54 L=10,120,10
   54   SCALE(L/10)=1.0/((((J-1)*120)+L)/10.0**5)
        WRITE(5,2004) SCALE
 2004   FORMAT(2X,12F10.0)
        WRITE(5,2005)
 2005   FORMAT(2X,'NO. OF OCCUR OF EACH FREQ FOR 20 PITCH PER'
       1   2X'NORMALIZED TO 300.'//)
   55   CONTINUE
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
C       FIND OUT WHICH MASTER THIS IS AND STORE THE HISTOGRAM
C       AND CHANNEL FIRINGS.

  201   WRITE (6,1070)
 1070   FORMAT('  WHAT IS THE CODE OF THIS SOUND?'/)
        READ (5,1060) PR
 1060   FORMAT(1A2)
        HH=0
        DO 202 J=1,25
        M=J
        IF(PR.EQ.SOUND(J)) GO TO 204
  202   CONTINUE

C       IF THIS IS AN HH, FIND OUT WHICH ONE.

        IF(PR.NE.'HH') GO TO 340
        WRITE(6,1085)
 1085   FORMAT('   WHAT SOUND GOES WITH THE HH'/)
        READ (5,1060) PR
        HH=11
        IF(PR.EQ.'NW') GO TO 204
        DO 320 J=1,10
        HH=J
        IF(PR.EQ.SOUND(J)) GO TO 204
  320   CONTINUE
  340   WRITE (6,1090) PR
 1090   FORMAT(5X,1A2,'  IS NOT A VALID CODE!'/)
        GO TO 201
  204   DO 206 J=1,224
  206   DAT(J)=PATTSN(J)
        DO 218 J=1,32
  218   DAT(J+224)=CHANEL(J)
        DAT(225)=STATSQ
        IF(HH.LE.0) WRITE(7'M) DAT
        IF(HH.GT.0) WRITE(3'HH)  DAT
        RAWMOM=0.0
        READ (7'28) DAT

C       CALCULATE MOMENTS AND STORE RESULTS.

        DO 220 K=1,100
  220   RAWMOM=RAWMOM-(101-K)*STATS(K)
        DO 230 K=1,230
  230   RAWMOM=RAWMOM+K*STATS(K+100)
        TEMPMO=0
        DO 240 K=331,480
  240   TEMPMO=TEMPMO+STATS(K)
        RAWMOM=RAWMOM+230.0*TEMPMO
        DO 250 J=1,3
        MOM(J)=0.0
        EN=STP(J)
        DO 250 K=1,EN
        KK=OFSET(J)+K
  250   MOM(J)=MOM(J)+K*STATS(KK)
        MOM(3)=MOM(3)+210.0*TEMPMO
        IF(RAWMOM.GT.20000.0) RAWMOM=20000.0
        INMOM(1)=RAWMOM/10.0
        DO 260 K=1,3
  260   INMOM(K+1)=MOM(K)/10.0
        KK=(M-1)*4
        IF(HH.GT.0) KK=(HH-1)*4+100
        DO 270 K=1,4
  270   DAT(KK+K)=INMOM(K)
        WRITE (7'28) DAT
        STOP
        END
```

VITA

Donald Bruce Warmuth was born 5 February 1946 in Cleveland, Ohio. He gratuated from high school in Toledo, Ohio in 1964 and attended The University of Michigan from which he received the degree of Bachelor of Electrical Engineering in December 1968. Upon graduation, he received a commission in the USAF through the ROTC program. He served as a project engineer and test director at the Electronic Systems Division of the Air Force Systems Command at L.G. Hanscom Field, Massachusetts, until September 1972. He then served as the Chief of Maintenance in th 2015 Communications Squadron, Randolph AFB, Texas,until entering the School of Engineering, Air Force Institute of Technology, (AFIT), in May 1974. He received the degree of Master of Science, Electrical Engineering, from AFIT in December 1975 and immediately entered the PhD program at AFIT.

Permant address: 30615 Hunters Lane
Farmington, Michigan 48024

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/DS/EE/78-3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AUTOMATIC RECOGNITION OF SYNTHETIC SPEECH USING AN ELECTRONIC MODEL OF THE MIDDLE AND INNER EAR		5. TYPE OF REPORT & PERIOD COVERED PhD Dissertation
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald B. Warmuth		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit 72330337
11. CONTROLLING OFFICE NAME AND ADDRESS Aerospace Medical Research Laboratories (BB) Aerospace Medical Division (AFSC) Wright-Patterson AFB, Ohio 45433		12. REPORT DATE 5 June 1978
		13. NUMBER OF PAGES 164
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

Jerral F. Guess, Capt, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech Recognition
Speech Understanding
Speech Analysis System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A phoneme-based automatic speech recognition system was developed and tested using synthetic speech. The acoustic signal is divided into short segments for analysis; segments are either a single pitch period of voiced speech or a 10 ms sample of voiceless speech. These segments are independently analyzed and given a phonemic name by three different measures. The sub-phonemic segments are grouped using measures which reflect dynamic changes in the speech signal. next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 (Con't)

Each group of segments represents a phoneme and is identified by simple algorithms operating on the string of phonemically-named segments that form the group.

The phoneme-based recognition system was tested using isolated synthesized words which permitted evaluation with connected strings of phonemes but stopped short of requiring development of word boundary rules. The tests consisted of 100 phonemically balanced words containing 281 phonemes. Of these, 245 were correctly identified, 23 were mis-identified, 13 were missed entirely, and 11 were added. However, many of these errors were predictable or understandable and may be overcome at a higher (word or phrase) level.